1997-03

Modeling and analysis of helicopter ground resonance utilizing symbolic processing and dynamic simulation software

Robinson, Christopher S

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

MODELING AND ANALYSIS OF HELICOPTER GROUND RESONANCE UTILIZING SYMBOLIC PROCESSING AND DYNAMIC SIMULATION SOFTWARE

by

Christopher S. Robinson

March, 1997

Thesis Advisor: E. Roberts Wood

Approved for public release; distribution is unlimited

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5101

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1997	Engineer's Thesis

4. TITLE AND SUBTITLE
Modeling and Analysis of Helicopter Ground Resonance Utilizing Symbolic Processing and Dynamic Simulation Software

5. FUNDING NUMBERS

6. AUTHOR(S)
Robinson, Christopher S.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis develops a technique for formulating the full nonlinear equations of motion for a coupled rotor-fuselage system utilizing the symbolic processing software MAPLE®. The symbolic software is further utilized to automatically convert the equations of motion into C, Fortran or MATLAB® source code formatted specifically for numerical integration. The compiled source code can be accessed and numerically integrated by the dynamic simulation software SIMULINK®. SIMULINK® is utilized to generate time history plots of blade and fuselage motion. These time traces can be used to explore the effects of damping nonlinearities, structural nonlinearities, active control, individual blade control, and damper failure on ground resonance. In addition, a MATLAB® program was developed to apply the Moving Block Technique for determining modal damping of the rotor-fuselage system from the time marching solutions.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Helicopter, ground resonance, air resonance, mechanical instability, active control, dynamics, MAPLE®, MATLAB®, SIMULINK®, numerical simulation, moving block, nonlinear	138
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Approved for public release; distribution is unlimited

MODELING AND ANALYSIS OF HELICOPTER GROUND RESONANCE UTILIZING SYMBOLIC PROCESSING AND DYNAMIC SIMULATION SOFTWARE

Christopher S. Robinson
Lieutenant United States Navy
B.S., Rensselaer Polytechnic Institute

Submitted in partial fulfillment of the
requirements for the degree of

AERONAUTICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL

March 1997

ABSTRACT

This thesis develops a technique for formulating the full nonlinear equations of motion for a coupled rotor-fuselage system utilizing the symbolic processing software MAPLE®. The symbolic software is further utilized to automatically convert the equations of motion into C, Fortran or MATLAB® source code formatted specifically for numerical integration. The compiled source code can be accessed and numerically integrated by the dynamic simulation software SIMULINK®. SIMULINK® is utilized to generate time history plots of blade and fuselage motion. These time traces can be used to explore the effects of damping nonlinearities, structural nonlinearities, active control, individual blade control, and damper failure on ground resonance. In addition, a MATLAB® program was developed to apply the Moving Block Technique for determining modal damping of the rotor-fuselage system from the time marching solutions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS, ACRONYMS AND/OR ABBREVIATIONS

Variables

$\vec{\rho}$	Position vector
$\vec{A}$	Inertia matrix
$\vec{f}$	Vector containing spring, damping, nonlinear and generalized force terms
ς	Damping ratio
ζ	Lead-Lag displacement of rotor blade
β	Flap displacement of rotor blade
ω	Frequency
ϕ	Active swashplate control phase angle
λ	Rotor inflow ratio
μ	Rotor advance ratio
Ω_0	Nominal frequency = fuselage natural frequency in x-direction
θ_c	Lateral cyclic for swashplate control input
ψ_k	Azimuth position of k^{th} rotor blade
Φ_k	Azimuth phase angle of k^{th} rotor blade
θ_k	Active control pitch input for k^{th} rotor blade
ω_p	Pylon natural frequency for isotropic case
θ_s	Longitudinal cyclic for swashplate control input
C_β	Linear damping coefficient in flap
C_ζ	Linear damping coefficient in lead-lag
c_1 , c_2	Same as c_x, c_y
C_1 , C_2	Fuselage linear damping constants (translational)
c_{d0}	Rotor blade parasite drag coefficient
C_p	Pylon damping for isotropic case
CR_1 , CR_2	Fuselage linear damping constants (rotational)
c_x , c_y	Linear damping coefficients for fuselage translation
D	Dissipation function
dF_y	Differential force on blade element in the direction of blade deformed y-axis
dF_z	Differential force on blade element in the direction of blade deformed z-axis
e	Rotor hinge offset
$F(\omega)$	Fourier transform of f(t)
F_i	Generalized force
h	Distance between origin of the hub coordinate system and origin of the fuselage coordinate system
$H(\tau)$	Heaviside step function
I_{11} , I_{22} , I_{12}	Fuselage rotational inertias and product of inertia
K_β	Elastic stiffness in flap
K_ζ	Elastic stiffness in lead-lag

k, k_b	Wave number of moving block frequency
K_1, K_2	Fuselage stiffness constants (translational)
K_d	Duffing spring constant
K_p	Pylon effective stiffness for isotropic case
KR_1, KR_2	Fuselage constants (rotational)
K_s	Stiffness used to simulate rotor blade lead-lag stops
M_β	Aerodynamic moment about rotor blade hinge in flap direction
M_ζ	Aerodynamic moment about rotor blade hinge in lead-lag direction
m'	Rotor blade mass per unit span
M_1, M_2	Effective fuselage mass in 1 and 2 directions (x and y)
m_b	Mass of rotor blade
M_p	Pylon effective mass for isotropic case
M_x, M_y	Same as M_1, M_2
N	Number of rotor blades; number of points in time history signal (signal length)
N_b	Number of signal points in moving block (block length)
q_i	Generalized displacement
r	Distance from blade hinge (or effective hinge) to an arbitrary point on a rotor blade elastic axis
r_1, r_2	Fuselage pitch and roll displacements
R_{cg}	Distance from hinge offset to center of gravity of a rotor blade
T	Kinetic energy
t	time
T_1, T_2, T_3	Coordinate transformation matrices
U	Potential energy
u_1, u_2	Fuselage displacements in the 1 and 2 directions (x and y)
U_p	Perpendicular relative air velocity (perpendicular to blade span axis and parallel to blade deformed z-axis)
U_R	Radial relative air velocity (parallel to blade span axis)
U_T	Tangential relative air velocity (perpendicular to blade span axis and parallel to blade deformed y-axis)
V_1, V_2	Nonlinear damping constants
V_x, V_y	Nonlinear damping constants
z	Blade stop angle

Superscripts, Subscripts and Acronyms

B	Refers to rotor blades
Bd	Blade deformed coordinate system
Bu	Blade undeformed coordinate system
cg	Center of gravity
F	Fuselage coordinate system, refers to fuselage
FFT	Fast Fourier Transform
H	Hub coordinate system
I	Inertial coordinate system

i	Refers to i^{th} generalized displacement or force in Lagrangian derivation
k	Refers to k^{th} rotor blade
LQG	Linear quadratic Gaussian
LQR	Linear quadratic Regulator
P	Refers to a point on the elastic axis of a rotor blade
rot	Pertaining to fuselage rotational degrees of freedom
T	Matrix transpose
trans	Pertaining to fuselage translational degrees of freedom

ACKNOWLEDGEMENT

I dedicate this work to my wonderful wife, Christine, and our beautiful daughter, Mary Elizabeth. Special thanks to Professor Bob Wood for his enthusiasm and guidance throughout this project.

I. INTRODUCTION

The motivation for this study grew from interest in smart materials and their application to helicopter dynamic problems. Smart materials can be used for aeroelastic tailoring of rotor system dynamic components and also as a mechanism for implementation of active rotor control without adding unwanted mechanical complexities. Active control of helicopter rotor blades has several potential benefits. Among these benefits are reductions in vibration and acoustic signature as well as elimination of mechanical and aeromechanical instabilities such as air and ground resonance. The idea of active control of helicopter rotors has received much attention in recent years. One of the driving forces behind this interest is the increased popularity of bearingless rotors, which offer the benefits of simplistic design and maintainability, but often give rise to air and ground mechanical stability problems.

It was immediately apparent that in order to conduct an in-depth investigation into the application of smart material technology to aeroelasticity and helicopter active control, the ability to accurately model coupled rotor-fuselage dynamics was needed. Several software packages such as CAMRAD® by Johnson Aeronautics, FLIGHTLAB® by Advanced Rotorcraft Technologies Corporation, and UMARC®, developed at the University of Maryland, offer this capability. All of these codes, while very capable, are quite complex and require considerable experience in order for a user to become proficient in applying them. As a result, this study was initiated in order to develop a computational tool for modeling and analysis of coupled rotor fuselage dynamics by utilizing readily available and generally applicable technical and mathematical software. The scope of this study involves utilizing the symbolic manipulation software, MAPLE ® by Waterloo

Software, and the dynamic simulation software, SIMULINK® by The Mathworks, to model and simulate the unstable mechanical phenomenon of helicopter ground resonance. The resulting models will be used to do parametric studies of ground resonance, including a look at the effect of active rotor control using fuselage state feedback. In order to better quantify the effects of parameter variations and active control schemes from simulation data, the moving block technique was utilized for time history analysis to quantify modal damping characteristics.

Ground resonance is a potentially destructive mechanical instability that can occur in helicopters with fully articulated, bearingless, or hingeless main rotor designs. The phenomenon of ground resonance is the result of a coupling between fuselage motion on its landing gear and rotor blade lead-lag and flap motion. The equations of motion describing the coupled rotor-fuselage system are nonlinear and generally quite complex even for simplified models. Procedures and techniques for dealing with the various complexities of ground resonance and other mechanical and aeromechanical phenomena that are characteristic of helicopters have been extensively investigated over the past few decades and an abundance of literature is available on the subject. The following section presents a brief overview of various approaches and techniques used in modeling coupled rotor-fuselage systems for the purpose of studying the ground resonance problem and active rotor control.

II. BACKGROUND

The following paragraphs discuss the phenomenon of ground resonance and some of the derivation and modeling techniques utilized through the years to manage the formulation and analysis of the complex equations of motion.

Ground resonance has been an observed happening in rotorcraft since the first autogyros were flown early in the 20th century. It can occur when any rotor system is placed on a flexible support. Essentially, a perturbation of a rotor blade causes the rotor center of gravity to shift creating an inertial load on the fuselage. The fuselage is flexibly connected to the ground via its landing gear and will start to oscillate in response to this inertial load. For a certain range of rotor rotational speeds, the fuselage oscillations will cause the amplitude of the blade oscillations about their respective hinges to increase, and this further increases the inertial forces on the fuselage. If left to its own accord, this coupling of fuselage and rotor blade motion will increase in amplitude until some nonlinear restoring force brings the system into a limit cycle or until some part of the rotorcraft fails [Ref. 1].

No discussion of helicopter ground resonance would be complete without considering the classic work of Coleman and Feingold [Ref. 2] completed in the 1940's. Coleman and Feingold successfully analyzed the coupling between a rotor and fuselage and identified the ground resonance instability as a purely self excited, elasto-mechanical phenomenon. Their study was based on a simplified three-bladed rotor model which is the basis for one of the models considered in this study. The model allows for hub translational degrees of freedom in one plane and rotor blade lead-lag degrees of freedom in the same plane (see Figure 3.1). In deriving the equations of motion, Coleman makes

use of "Coleman" coordinates [Ref. 2] and complex variables to reduce the number of equations that describe the system to two complex (four real). From the equations of motion, the characteristic equation is derived by assuming a solution that has the rotor center of mass and fuselage center of mass moving in an elliptic whirling motion. The roots of the characteristic equation are the characteristic whirling speeds of the rotor , and the nature of these roots indicate the nature of the system stability, i.e., whether or not the rotor rotational speed resides in the self excited region. Coleman and Feingold ultimately reduce the results of their study to a series of graphs which can be applied to a wide range of rotor configurations.

Coleman and Feingold's work became the basis for the evolution of theory and design techniques used for dealing with ground resonance. Although this classic theory offers much insight and understanding into the phenomenon, especially for conventional articulated rotor systems, the increasing popularity of hingeless and bearingless rotor designs in modern helicopters and the growing desire to eliminate the need for mechanical dampers requires more sophisticated analytical techniques.

As computational power improved with the evolution of digital computers, more general techniques for analyzing rotor system stability came into being. Peters and Hohenemser [Ref. 3] apply Floquet analysis to the problem of lifting rotor stability. Floquet analysis is a method which can be used to determine the stability of solutions to systems of linear ordinary differential equations with periodic coefficients. The Floquet transition matrix which relates the system state variables at the beginning and end of a rotational period is computed by numerical time wise integration. The eigenvalues of the transition matrix are a measure of system stability. Hammond [Ref. 4] applies Floquet

analysis to the prediction of mechanical instabilities, specifically examining the case of unbalanced lead-lag damping. The unbalanced problem requires solution of the equations of motion with the blade equations expressed in the rotating reference frame because a transformation to the fixed system is no longer possible for a ground resonance analysis as was possible for the isometric case. As a result, you are left with a system of equations with periodic coefficients which can be handled by the Floquet method.

Hingeless and bearingless rotor configurations often face the additional difficulty of air resonance. Aerodynamics may play more than a passive roll in the ground resonance regime in hingeless systems in contrast to articulated systems where aerodynamics have little effect. As a result, more complex models are required to accurately represent the physics of the helicopter aeromechanical stability problem. Models must include blade flap and torsional degrees of freedom as well as lead-lag degrees of freedom. Fuselage models also should include pitch and roll as well as translational degrees of freedom. Aerodynamic models can range from quasi-steady strip theory to unsteady aerodynamic theories which include elaborate wake models or dynamic inflow models. Ormiston [Ref. 5] utilizes a rigid blade and rigid fuselage model with flap-lag and pitch-roll degrees of freedom to conduct parametric investigations based on an eigenvalue analysis. As is typical, the equations of motion were derived by a Newtonian approach and the resulting system of nonlinear differential equations are linearized for small perturbations. The model includes linear rotor blade and landing gear springs, viscous damping, and quasi-steady aerodynamics. Freidmann and Venkatesan [Ref. 6] and Freidmann and Warmbrodt [Ref. 7] derive the complete set of governing equations of a helicopter rotor coupled to a rigid body fuselage. The equations account for rotor blade

elastic deformations and include quasi-steady aerodynamics or modified Theordorsen unsteady aerodynamic theory. In deriving the full equations of motion, Freidmann et al., stress the importance of applying an ordering scheme to the process in order to handle the complexity of the equations and enormous number of terms generated by their expansion. The equations, as presented by Freidmann et al. [Ref. 6 and 7], are in a form which makes them generally applicable to a wide range of rotorcraft problems.

Another interest in the study of helicopter ground resonance is the effect that nonlinear elastic and damping forces have on stability. Tongue [Ref. 1], Tongue and Flowers [Ref. 8 and 9], Tongue and Jankowski [Ref. 10], and Tang and Dowell [Ref. 11], use variations of the nonlinear technique of harmonic balance using describing functions to represent nonlinear damping. The technique is useful for investigating limit cycle behavior of strongly nonlinear systems and its impact on system stability.

Active control of rotor systems and its application to stabilizing ground and air resonance has been investigated by Straub [Ref. 12] and Straub and Wambrodt [Ref. 13]. In both of these studies the nonlinear periodic equations of motion, derived with a method similar to that of Freidmann and Venkatesan [Ref. 6], are linearized and incorporated into a state space model in which active control inputs are input to the rotor blades from the fixed coordinate system via a swashplate. The state space model is then used to study the influence that state feedback gain and phase have on system damping.

Helicopter aeromechanical instabilities can be analyzed by methods ranging from Coleman's classic analysis to direct time integration of the equations of motion. As engineers strive to develop rotor systems free of ground and air resonance which do not require the addition of maintenance intensive mechanical damping systems, more

elaborate models will be needed to accurately capture all physical aspects of the problem. To achieve the truly damperless rotor Ormiston [Ref. 14] addresses three different approaches which may be feasible, 1) incorporating high damping material into the blade or flexbeam structure, 2) automatic feedback control, and 3) development of aeroelastic couplings to provide inherent stability. These three approaches have provided the impetus behind the work which follows. The goals of the following study were to develop a modeling technique utilizing symbolic processing to manage the complexity of deriving and coding the coupled rotor-fuselage equations of motion, incorporate the resulting model into a dynamic simulation environment, and have a final product which provides a useful tool for conducting parametric studies of helicopter aeromechanical behavior.

III. EQUATIONS OF MOTION

The equations of motion for a coupled rotor-fuselage system were derived using the Lagrangian approach. This study was concerned with two models, one being characterized as simple and the other as complex. The Lagrangian approach was selected as opposed to the Newtonian because it is easily implemented with the aid of a symbolic processor. For a historical note, it is interesting that Lagrange himself recognized the suitability of his methods to routine processing. He states in his *Méchanique Analytique*,

> The methods which I present here do not require either constructions or reasonings of geometrical or mechanical nature, but only algebraic operations proceeding after a regular and uniform plan. Those who love the Analysis, will see with pleasure Mechanics made a branch of it and will be grateful to me for having thus extended its domain.

The equations for both models considered were formulated with Lagranges method in their full nonlinear forms, i.e., no ordering scheme, small angle assumptions, or linearization techniques were applied during derivation and subsequent coding.

A. SIMPLIFIED MODEL DESCRIPTION

The simplified model is based on that used by Coleman [Ref. 2] , and is shown schematically in Figure 3.1. A three bladed model will be the only case considered in this report, but all mathematical modeling methods used in this study can be easily generalized to any number of blades. Elastic forces generated by rotor blade and flexbeam motion were modeled as a linear torsional spring located at the effective hinge position of the blade. The landing gear stiffness was also modeled with linear springs. For the basic simplified model, landing gear and lead-lag damping was modeled with linear dashpot type dampers. Addition of nonlinear mechanical effects such as hardening and softening springs, hydraulic damping, and lead-lag stops, will be discussed in a later section.

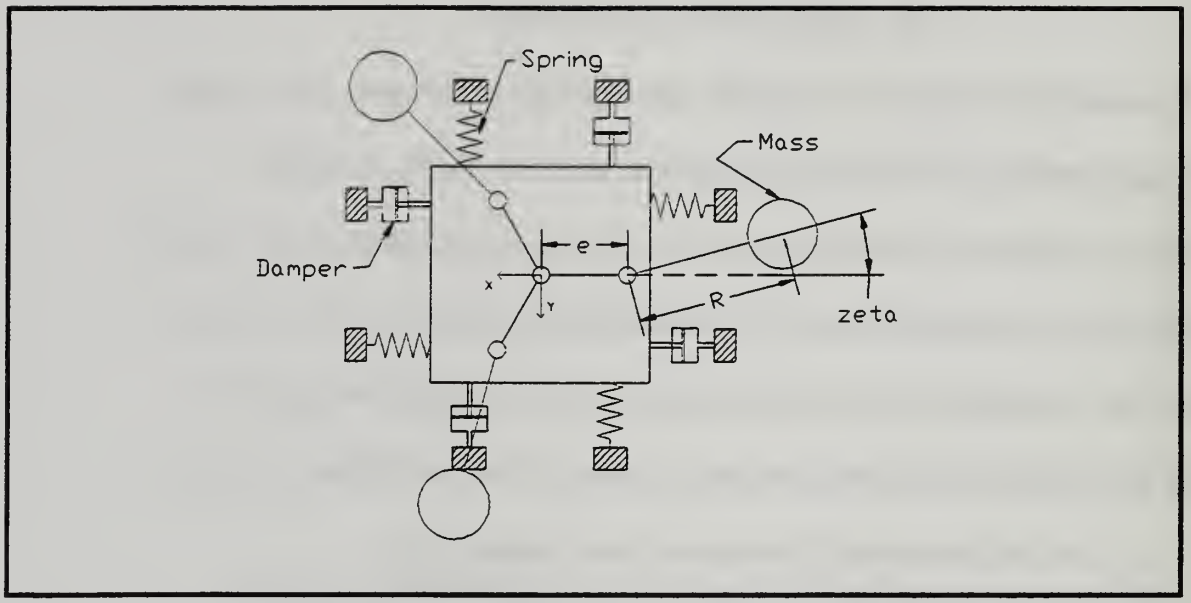


Figure 3. 1 Schematic of Simplified Rotor-Fuselage System

This model allows for the following degrees of freedom:

u_1 = Fuselage translation in 1-direction (x-direction).

u_2 = Fuselage translation in 2-direction (y-direction).

ζ_k = Lead-lag angular displacement of k^{th} rotor blade.

A. COMPLEX MODEL DESCRIPTION

The complex model is based on that used by Straub [Ref. 12]. This model assumes rigid blades and fuselage. The blade and flexbeam elastic forces are modeled, as in the simplified model, as equivalent torsional springs located at effective hinge positions offset from the rotor hub (flap and lead-lag hinges are assumed to be coincident).

This model allows for the following degrees of freedom:

u_1 = Fuselage translation in 1-direction (x-direction).

u_2 = Fuselage translation in 2-direction (y-direction).

r_1 = Fuselage rotation about 1-axis (roll).

r_2 = Fuselage rotation about 2-axis (pitch).

ζ_k = Lead-lag angular displacement of k^{th} rotor blade.

β_k = Flap angular displacement of k^{th} rotor blade.

C. COORDINATE SYSTEMS AND TRANSFORMATIONS

In developing the equations of motion for the two coupled rotor-fuselage models five coordinate systems were utilized with transformations between the various systems based on Euler angle rotations. The five coordinate systems are (1) inertial, fixed relative to the Earth, (2) fuselage, fixed to center of gravity of fuselage, (3) hub, parallel to fuselage system but offset a distance h in the positive z (or 3) direction, (4) undeformed blade, fixed to the effective hinge position on the k^{th} blade, (5) deformed blade, fixed to the effective hinge position on the k^{th} blade, but with the x-axis coincident with the blade 'elastic' axis. Table 3.1 summarizes the notation used for the various coordinate systems.

Table 3.1 Coordinate System Notation

Coordinate System	Representative Notation for a Vector in this System
Inertial	$\begin{bmatrix} x & y & z \end{bmatrix}^T$
Fuselage	$\begin{bmatrix} \tilde{x} & \tilde{y} & \tilde{z} \end{bmatrix}^T$
Hub	$\begin{bmatrix} \bar{x} & \bar{y} & \bar{z} \end{bmatrix}^T$
Undeformed blade	$\begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \end{bmatrix}^T$

Coordinate System	Representative Notation for a Vector in this System
Deformed blade	$\begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \end{bmatrix}^T$

The following generic transformations are defined:

$$T_1(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{3.1}$$

$$T_2(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \tag{3.2}$$

$$T_3(\alpha) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Where in general, T_1, T_2, and T_3 are rotations about the 1, 2, and 3 axes respectively . This notation can be directly utilized with the symbolic processor and will be used in the following section where the energy expressions necessary for the Lagrangian derivation are defined. The order of fuselage rotations when using these Euler angle transformations will be pitch - roll, and the order of rotor blade angular displacements will be flap-lag. The following relations summarize the coordinate transformations used for the simplified and complex models:

Simplified model:

1. Hub to Inertial:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = I \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} \quad ; \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad ; \quad \textit{Systems are parallel} \tag{3.4}$$

2. **Blade undeformed to Hub:**

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \left[T_3(\psi_k) \right]^T \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} \tag{3.5}$$

3. **Blade deformed to Blade undeformed:**

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \left[T_3(\zeta_k) \right]^T \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} \tag{3.6}$$

<u>Complex model:</u>

1. **Fuselage to Inertial:**

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \left[T_1(r_1) T_2(r_2) \right]^T \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} \tag{3.7}$$

2. **Hub to Fuselage:**

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = I \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} \quad ; \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad ; \quad \textit{Systems are parallel.} \tag{3.8}$$

3. **Blade undeformed to Hub:**

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \left[T_3(\psi_k)\right]^T \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} \tag{3.9}$$

4. Blade deformed to blade undeformed:

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \left[T_3(\zeta_k)T_2(\beta_k)\right]^T \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} \tag{3.10}$$

D. DERIVATION UTILIZING SYMBOLIC PROCESSOR

This section summarizes the development of the energy expressions necessary for the Lagrangian derivation. Here, the equations for the complex model are developed to illustrate how the symbolic processor was utilized.

The Langrange equation can be expressed as follows:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} + \frac{\partial D}{\partial \dot{q}_i} = F_i \tag{3.11}$$

Where, T is the kinetic energy, U is the potential energy, D is the dissipation function, F_i is a generalized force, and q_i is a generalized displacement. The generalized force term, F_i, will describe the aerodynamic forces on the individual rotor blades and will be discussed in a later section, as a result, this derivation develops only the system of homogeneous equations. The various energy terms can be broken down into two categories, terms due to blade motion and terms due to fuselage motion, to give the following equations:

$$T = T_F + \sum_{k=1}^{N} \left(T_B\right)_k \tag{3.12}$$

$$U = U_F + \sum_{k=1}^{N} (U_B)_k \tag{3.13}$$

$$D = D_F + \sum_{k=1}^{N} (U_B)_k \tag{3.14}$$

Where the subscripts F and B indicate fuselage and rotor blade respectively.

The kinetic energy of the k^{th} rotor blade is given by the following expression:

$$(T_B)_k = \int_0^R \frac{1}{2} m' \left(\dot{\vec{\rho}} \cdot \dot{\vec{\rho}} \right) dr \tag{3.15}$$

Where $\vec{\rho}$ is the position of a point on the elastic axis of the k^{th} rotor blade with respect to the inertial coordinate system at any instant in time, and m' is the mass distribution per unit length of the blade (for this study mass distribution per unit length is assumed to be uniform). The position of a point on the elastic axis of a rotor blade, $\vec{\rho}$, is expressed as the sum of relative positions with respect to the various coordinate systems transformed to the inertial system. Thus,

$$\vec{\rho} = \left(\vec{\rho}_{F\_I}\right)_I + \left(\vec{\rho}_{H\_F}\right)_I + \left(\vec{\rho}_{Bu\_H}\right)_I + \left(\vec{\rho}_{Bd\_Bu}\right)_I + \left(\vec{\rho}_{P\_Bd}\right)_I , \tag{3.16}$$

where, for example, the term $\left(\vec{\rho}_{Bu\_H}\right)_I$ is the position of the origin of the undeformed blade coordinate system with respect to the hub coordinate system transformed into the inertial coordinate system. The individual terms of equation (3.16), referring to equations (3.1) through (3.3) are defined as follows:

$$\left(\vec{\rho}_{F\_I}\right)_I = \vec{\rho}_{F\_I} \tag{3.17}$$

$$\left(\vec{\rho}_{H\_F}\right)_I = \left[T_1(r_1)T_2(r_2)\right]^T \vec{\rho}_{H\_F} \tag{3.18}$$

$$\left(\vec{\rho}_{Bu\_H}\right) = \left[T_1(r_1)T_2(r_2)\right]^T \left[T_3(\psi_k)\right]^T \vec{\rho}_{Bu\_H} \tag{3.19}$$

$$\left(\vec{\rho}_{Bd\_Bu}\right)_I = \vec{\rho}_{Bd\_Bu} = \vec{0} \quad \textit{Origins are coincident} \tag{3.20}$$

$$\left(\vec{\rho}_{P\_Bd}\right)_I = \left[T_1(r_1)T_2(r_2)\right]^T \left[T_3(\psi_k)\right]^T \left[T_3(\zeta_k)T_2(\beta_k)\right]^T \vec{\rho}_{P\_Bd} \tag{3.21}$$

Where,

$$\vec{\rho}_{F\_I} = \begin{bmatrix} u_1 \\ u_2 \\ 0 \end{bmatrix} \;;\; \vec{\rho}_{H\_F} = \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix} \;;\; \vec{\rho}_{Bu\_H} = \begin{bmatrix} e \\ 0 \\ 0 \end{bmatrix} \;;\; \vec{\rho}_{P\_Bd} = \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} \tag{3.22}$$

Equations (3.22) are substituted into the equations of (3.17) through (3.21) and the matrix multiplication is performed with the results substituted into equation (3.16). This gives a vector expression for the position of an arbitrary point on the elastic axis of the k^{th} rotor blade with respect to the inertial coordinate system at any instant in time in terms of the system degrees of freedom. The time derivative of this expression gives the velocity, $\dot{\vec{\rho}}$, which is substituted into equation (3.15) to give the kinetic energy for k^{th} rotor blade. All of the calculus and algebra was accomplished with MAPLE® (see Appendix A for a look at the MAPLE® worksheet).

The elastic forces generated by rotor blade motion give rise to a potential energy term in the Lagrange equation. Since a rigid blade model was assumed, the potential

energy was modeled using equivalent torsional springs to restrain the rotor blade, with spring constants selected to approximate elastic forces due to in plane and out of plane bending of the rotor blade (and the flexbeam in the hingeless case). The potential energy of the k^{th} rotor blade is

$$\left(U_B\right)_k = \frac{1}{2}K_\beta \beta_k^{\,2} + \frac{1}{2}K_\zeta \zeta_k^{\,2} \tag{3.23}$$

An explanation of the validity of using an equivalent torsional spring system to model the elastic forces of a deformed rotor blade is given in some detail by Venkatesan and Friedmann [Ref. 6].

System damping is modeled in energy form by use of a dissipation function, which for the k^{th} rotor blade of the complex rotor model is

$$\left(D_B\right)_k = \frac{1}{2}C_\beta \dot{\beta}_k^{\,2} + \frac{1}{2}C_\zeta \dot{\zeta}_k^{\,2} \tag{3.24}$$

For the fuselage, the kinetic energy in terms of translational and rotational degrees of freedom is

$$\left(T_F\right)_{trans} = \frac{1}{2}M_1 \dot{u}_1^{\,2} + \frac{1}{2}M_2 \dot{u}_2^{\,2} \tag{3.25}$$

$$\left(T_F\right)_{rot} = \frac{1}{2}I_{11}\dot{r}_1^{\,2} + \frac{1}{2}I_{22}\dot{r}_2^{\,2} - 2I_{12}\dot{r}_1\dot{r}_2 \tag{3.26}$$

The fuselage potential energy is

$$\left(U_F\right)_{trans} = \frac{1}{2}K_1 u_1^{\,2} + \frac{1}{2}K_2 u_2^{\,2} \tag{3.27}$$

$$\left(U_F\right)_{rot} = \frac{1}{2}KR_1 r_1^{\,2} + \frac{1}{2}KR_2 r_2^{\,2} \tag{3.28}$$

The dissipation functions for the fuselage are

$$\left(D_F\right)_{trans} = \frac{1}{2}C_1\dot{u}_1^{\,2} + \frac{1}{2}C_2\dot{u}_2^{\,2} \tag{3.29}$$

$$\left(D_F\right)_{rot} = \frac{1}{2}CR_1\dot{r}_1^{\,2} + \frac{1}{2}CR_2\dot{r}_2^{\,2} \tag{3.30}$$

The resulting inputs for the fuselage terms in equations (3.12) through (3.14) are

$$T_F = \left(T_F\right)_{trans} + \left(T_F\right)_{rot} \tag{3.31}$$

$$U_F = \left(U_F\right)_{trans} + \left(U_F\right)_{rot} \tag{3.32}$$

$$D_F = \left(D_F\right)_{trans} + \left(D_F\right)_{rot} \tag{3.33}$$

All of the energy expressions defined above were entered into a MAPLE® worksheet programmed to apply equation (3.11) and generate the equations of motion corresponding to each of the system's degrees of freedom (see Appendix A). An important characteristic of MAPLE® is that it allows for distinction between dependent and independent variables via functional notation, e.g., to indicate a variable 'X' is a function of time (t) simply write it as 'X(t)'. It is also important to note that when applying Lagrange's equation in MAPLE®, derivatives are only understood when taken with respect to independent variables, so when taking derivatives with respect to the degrees of freedom and the time rates of change of the degrees of freedom, the time functional notation which represents these variables must be converted to independent variable notation. For example, the flap angle degree of freedom, $\beta(t)$, and its time rate of change, $\frac{\partial \beta(t)}{\partial t}$, would have to be replaced in all of the energy expressions by the independent variables, β and $d\beta$, respectively, in order for terms like $\frac{\partial T}{\partial q_i}$ and

$\frac{\partial T}{\partial \dot{q}_i}$ (where $q_i = \beta(t)$ and $\dot{q}_i = \frac{\partial \beta(t)}{\partial t}$) to be evaluated properly by the MAPLE® symbolic engine. Additionally, for the time derivative term, $\frac{\partial}{\partial t}\left(\frac{\partial T}{\partial \dot{q}_i}\right)$, to be evaluated properly, all degrees of freedom expressed in independent notation must be converted back to time dependent notation. The MAPLE® code which accomplishes the above manipulations for the complex model is contained in Appendix A.

The equations of motion generated by the MAPLE® program for the simplified model were verified against the equations used by Flowers and Tongue [Ref. 5]. Flowers and Tongue also utilized a Lagrangian approach and symbolic manipulation to arrive at the following equations of motion for a model similar to the one described in Figure 3.1.

$$M_x \ddot{x} + C_x \dot{x} + V_x \dot{x}|\dot{x}| + K_x x = m_b R \sum_{k=1}^{N}\left[\ddot{\zeta}_k \sin(\psi_k + \zeta_k) + (\Omega + \dot{\zeta}_k)^2 \cos(\psi_k + \zeta_k)\right] \tag{3.34}$$

$$M_y \ddot{y} + C_y \dot{y} + V_y \dot{y}|\dot{y}| + K_y y = -m_b R \sum_{k=1}^{N}\left[\ddot{\zeta}_k \sin(\psi_k + \zeta_k) - (\Omega + \zeta_k)^2 \sin(\psi_k + \zeta_k)\right] \tag{3.35}$$

$$m_b R^2 \ddot{\zeta}_k + C_k \dot{\zeta}_k + K_k \zeta_k + m_b \Omega^2 eR \sin(\zeta_k) = m_b R\left(\ddot{x}\sin(\psi_k + \zeta_k) - \ddot{y}\cos(\psi_k + \zeta_k)\right) \tag{3.36}$$

Flowers and Tongue included an additional term in equations (3.34) and (3.35) involving the product of first derivatives and their absolute values. These terms represent nonlinear damping. Inclusion of nonlinear (hydraulic) damping in the dissipation function for input into Lagrange's equation will be discussed in a later section.

Equations (3.34) through (3.35) were compared against the equations of motion generated by the MAPLE® program for a three bladed simple rotor fuselage model and were found to match exactly (except for the nonlinear damping terms which were not included initially). The MAPLE® equations for this case are shown in Appendix B.

IV. BUILDING THE SIMULATION MODEL

A. S-FUNCTIONS AND CODE GENERATION

Construction of the simulation model from the equations of motion was based on the structure of the SIMULINK® S-function. The S-function defines the dynamics of a model. It can be written in C or Fortran code or as a MATLAB® m-file (a mathematical programming language with similar syntax to Fortran). The structure of the S-function is generic so as to allow for a wide range of functionality when programming the dynamics of various systems. SIMULINK® accesses an S-function through its numerical integration routines. The routines make calls to the S-function for specific information, the type of information returned is dependent on the value of a flag variable sent by the integration routine. For example,

flag = 0 S-function returns sizes of parameters and initial conditions ,

flag = 1 S-function returns state derivatives dx/dt,

flag = 3 S-function returns outputs.

The section of the S-function which computes the derivatives at each time step is the section which contains the equations of motion [Ref. 15].

As a result of the Lagrangian derivation , MAPLE® generated the equations of motion in the following form,

$$\vec{F}\left(\ddot{\vec{x}}, \dot{\vec{x}}, \vec{x}, t\right) = 0 \tag{4.1}$$

where $\vec{x}$ is a vector of displacement degrees of freedom of the system. Unfortunately this form is not very useful when it comes to programming a SIMULINK® S-function, so

MAPLE® was further used to manipulate the equations of motion into the following equivalent form,

$$\left[A\left(\dot{\vec{x}},\vec{x},t\right)\right]\ddot{\vec{x}} = \vec{f}\left(\dot{\vec{x}},\vec{x},t\right) \tag{4.2}$$

where A is an NxN matrix and $\vec{f}$ an Nx1 vector, with N = number of degrees of freedom of the system. This is possible since the equations are quasi-linear in the second derivative (acceleration) terms, i.e., no terms exist of types such as $\ddot{\vec{x}}^2$, or $\sin\left(\ddot{\vec{x}}\right)$, etc. This form can then be transformed from N second order equations to 2N first order equations as follows,

$$\begin{aligned} \dot{\vec{x}} &= \vec{w} \\ \dot{\vec{w}} &= [A]^{-1}\vec{f} \end{aligned} \tag{4.3}$$

These equations can be evaluated at each time step in a numerical simulation to give the state derivatives. The primary job to be accomplished with MAPLE® was to generate the expressions for the elements of $[A]$ and $\vec{f}$ (which can be quite lengthy) from the equations of motion. After this was accomplished, the MAPLE® code generation routine was used to automatically generate the optimized C or Fortran code that could be placed directly into an S-function template (See Appendix C for an example of the MAPLE® code generation results).

For each model, MAPLE® was used to generate the Fortran code representing the equations of motion. This code was used to create an S-function in MATLAB® m-file format. Because Fortran and m-file syntax are so similar, only minor editing was required. A copy of the S-function program for this case is contained in Appendix D.

B. NUMERICAL INTEGRATION ROUTINE (ODE SOLVER)

SIMULINK® provides several numerical ordinary differential equation (ode) solvers (numerical integration algorithms). The algorithms utilized in this study are from the Runge-Kutta (rk) family (rk45 and rk23). Runge-Kutta algorithms generally outperform other schemes for systems of nonlinear ordinary differential equations which are not too stiff. The rk algorithms also handle discontinuities well [Ref. 7].

For completeness and to describe how the integration algorithms interact with SIMULINK® S-functions, a brief description of the rk method is given. The first order ode (the following algorithm can be easily extended to systems of first order ode's),

$$\frac{dy}{dt} = f(y,t) \tag{4.4}$$

can be integrated between t_n and t_{n+1} to give the following,

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(y,t)\,dt \tag{4.5}$$

In the rk method, the integral expression on the right hand side of equation (4.5) is approximated with a numerical integration scheme, such as Simpson's 1/3 rule, resulting in the following expression for the value of y at the next time step,

$$y_{n+1} = y_n + \frac{h}{6}\left[f(y_n,t_n) + 4f(\bar{y}_{n+\frac{1}{2}}, t_{n+\frac{1}{2}}) + f(\bar{y}_{n+1}, t_{n+1}) \right] \tag{4.6}$$

The parameter h is the size of the time step, and $\bar{y}_{n+\frac{1}{2}}$ and $\bar{y}_{n+1}$ are initial estimates of y at the half and full step, and are given by the following expressions derived from Taylor expansions about y_n,

$$\bar{y}_{n+\frac{1}{2}} = y_n + \frac{h}{2} f(y_n, t_n) \tag{4.7}$$

$$\bar{y}_{n+1} = y_n + h\left[\theta f(y_n, t_n) + (1-\theta) f\left(\bar{y}_{n+\frac{1}{2}}, t_{n+\frac{1}{2}}\right)\right] \tag{4.8}$$

The variable θ is a weighting parameter that can be selected to optimize the accuracy of the numerical method [Ref. 16]. In SIMULINK®, it is the values of the function *f(y,t)* that are generated by the S-function when the integration routine calls with the proper flag. Thus, SIMULINK® provides the means of modeling any dynamic system for numerical simulation provided that the equations of motion are expressed as a system of first order ode's. It is important to note that the algorithm described above is only one variant of the rk family of ode solvers and that the actual routines utilized by SIMULINK® are somewhat more sophisticated.

Before beginning a simulation with SIMULINK®, the user sets several parameters which control the execution of integration routine. The user must designate a maximum and minimum (time) step size, simulation duration (start and stop time), and a tolerance which establishes the maximum relative error. If the algorithm cannot decrease the maximum relative error without going below the minimum step size, a warning message is displayed.

V. MODELING NONLINEAR EFFECTS

To include additional nonlinear terms into the overall model the equivalent energy expression representing the effect is simply added to the overall energy expression in the MAPLE® worksheet program. For example, to model a nonlinear flexbeam, a Duffing spring term can be added to the equation of motion of the kth rotor blade of the form

$$\left(K_d\right)_k \zeta(t)^3 \tag{5.1}$$

For the Lagrangian derivation, which the MAPLE® program performs, the equivalent potential energy term is given by

$$\int_0^{\zeta} \left(K_d\right)_k z^3 dz = \frac{1}{4}\left(K_d\right)_k \zeta^4 \tag{5.2}$$

This term is simply added to the expression representing the potential energy of the kth rotor blade. To add nonlinear damping to the rotor blades or fuselage a term of the following form is added to the respective dissipation function

$$V_x \dot{x}^2 |\dot{x}| \tag{5.3}$$

where V_x is the nonlinear hydraulic damping coefficient [Ref. 5].

The effect of lead-lag (flap) stops was also modeled by incorporating a simulated jump in lead-lag (flap) stiffness by use of Heaviside step functions in the potential energy expressions. For lead-lag stops the expression is

$$\frac{1}{2}K_s H(\zeta - z)(\zeta - z)^2 + \frac{1}{2}K_s(\zeta + z)^2 - \frac{1}{2}K_s H(\zeta + z)(\zeta + z)^2 \tag{5.4}$$

where the Heaviside step function, $H(\tau)$, is defined as

$$H(\tau) = \begin{cases} 0, & \tau < 0 \\ 1, & \tau \geq 0 \end{cases} \tag{5.5}$$

and z is the absolute displacement angle at which a rotor blade would engage the lead-lag (flap) stops, and K_s is the effective increase in lead-lag (flap) stiffness. For programming purposes, expression (5.4) can be written in terms of the signum( ) function as follows

$$\begin{aligned}&\frac{1}{4}K_s\zeta^2\,sigmum(\zeta-z)+\frac{1}{2}K_s\zeta^{\,2}-\frac{1}{4}K_sz^2\,sigmum(\zeta+z)-\frac{1}{2}K_sz\zeta\;sigmum(\zeta-z)\\&\quad+\frac{1}{4}K_sz^2\,sigmum(\zeta-z)-\frac{1}{2}K_sz\zeta\;sigmum(\zeta-z)-\frac{1}{4}K_s\zeta^{\,2}\;sigmum(\zeta+z)\\&\quad+\frac{1}{2}K_sz^2\end{aligned}\tag{5.6}$$

where

$$sigmum(x)=\frac{x}{|x|}\tag{5.7}$$

Thus, any structural or damping nonlinearity can be incorporated into the model by adding the appropriate energy expression into the MAPLE® worksheet, and then executing the worksheet to generate the updated code for incorporation into the SIMULINK® S-function.

VI. SIMULATION RESULTS

This section displays results of several simulations and demonstrates the unique capabilities and flexibility of the modeling method described in previous sections. Direct simulation allows analysis of any number of different configurations or scenarios, such as non-isotropic hub, one damper inoperative, or even simulated rotor blade damage. Though the time history plots in the following subsections do not indicate it, SIMULINK® offers the useful capability of being able to visualize the dynamics of a model as they progress, which can add valuable insight into the phenomenon being studied.

A. SIMPLE MODEL

The following table summarizes the parameters that can be set interactively for any simulation for the simple model. The table gives the representative nomenclature used to represent the parameter in the MAPLE® and S-function programs. The table is representative of a three bladed model.

Table 6.1 Simple Rotor Model Program Nomenclature

Parameter	As it Appears in MAPLE® and/or S-function code	Units
Rotor blade mass	mb(1), mb(2), mb(3)	mass
Fuselage effective mass in x and y direction	M(1), M(2)	mass
Distance from hinge to center of mass of blade	R	length
Rotor speed	Omega	rad/sec
Hinge offset	e1	length
Angle at which lead-lag stops engage	z	radians
Azimuth phase angle of rotor blade	Phi(1), Phi(2), Phi(3)	radians

Parameter	As it Appears in MAPLE® and/or S-function code	Units
Lead-lag linear damping coefficient	Czeta(1), Czeta(2), Czeta(3)	moment/ (rad/sec)
Lead-lag nonlinear damping coefficient	Vzeta(1), Vzeta(2), Vzeta(3)	moment/ $(rad/sec)^2$
Fuselage linear damping coefficient in x and y direction	c(1), c(2)	force/ (length/sec)
Fuselage nonlinear damping coefficient in x and y direction	v(1), v(2)	force/ $(length/sec)^2$
Lead-lag linear spring coefficient	Ke(1), Ke(2), Ke(3)	moment/rad
Lead-lag nonlinear spring coefficient (Duffing spring)	Kd(1), Kd(2), Kd(3)	moment/rad^3
Lead-lag stop spring coefficient	Ks(1), Ks(2), Ks(3)	moment/rad
Effective fuselage stiffness in x and y direction	K(1), K(2)	force/length
Fuselage states initial displacement conditions	xXi, xYi	length
Fuselage states initial rate conditions	xrXi, xrYi	length/sec
Blade states initial displacement conditions	x1i, x2i, x3i	rad
Blade states initial rate conditions	xr1i, xr2i, xr3i	rad/sec

Table 6.2 shows the basic simulation case for the parameters in table 6.1. These base values will serve as a starting point for each simulation, i.e. when a specific parameter is changed for a simulation, it is understood that all other parameters will be set to the base case values.

Table 6.2 Parameter Settings for Basic Simulation Case

mb(1)	mb(2)	mb(3)	M(1)	M(2)
0.1 slugs	0.1 slugs	0.1 slugs	6.5 slugs	6.5 slugs
R	Omega	e1	z	
10 ft	170 radians/sec	0.5 ft	$\pi/12$ radians	
Phi(1)	Phi(2)	Phi(3)		
0 radians	$2\pi/3$ radians	$4\pi/3$ radians		
c(1)	c(2)	v(1)	v(2)	
0 lbs/fps	0 lbs/fps	0 lbs/fps^2	0 lbs/fps^2	
Czeta(1)	Czeta(2)	Czeta(3)		
0 ft-lbs/(radians/sec)	0 ft-lbs/(radians/sec)	0 ft-lbs/(radians/sec)		
Vzeta(1)	Vzeta(2)	Vzeta(3)		
0 ft-lbs/(radians/sec)^2	0 ft-lbs/(radian/sec)^2	0 ft-lbs/(radian/sec)^2		
Ke(1)	Ke(2)	Ke(3)		
0 ft-lbs/radian	0 ft-lbs/radian	0 ft-lbs/radian		
Kd(1)	Kd(2)	Kd(3)		
0 ft-lbs/radian^3	0 ft-lbs/radian^3	0 ft-lbs/radian^3		
Ks(1)	Ks(2)	Ks(3)		
0 ft-lbs/radian	0 ft-lbs/radian	0 ft-lbs/radian		
K(1)	K(2)			
113,000 lbs/ft	113,000 lbs/ft			
xXi	xYi	x1i	x2i	x3i
0 ft	0 ft	0 radians	0 radians	0 radians
xrXi	xrYi	xr1i	xr2i	xr3i
0.5 ft/sec	0 ft/sec	0 radians/sec	0 radians/sec	0 radians/sec

The basic case is intentionally set up with zero damping and with a rotor speed set approximately at the center of the regressing lead-lag mode instability region. The first set of simulations will demonstrate the system behavior when excited with an initial fuselage velocity as indicated in Table 6.2. Figure 6.1 and Figure 6.2 show the lead-lag time

histories and the fuselage center of mass trajectory (displacements are in feet) for the basic case.

Figure 6.1 Rotor Lead-lag Displacements for Basic Parameter Case Settings, Center of Self Excited Region.

Unless otherwise specified, for plots of rotor blade motion, i.e., lead-lag and flap , the colors red, blue, and green distinguish the individual blades.

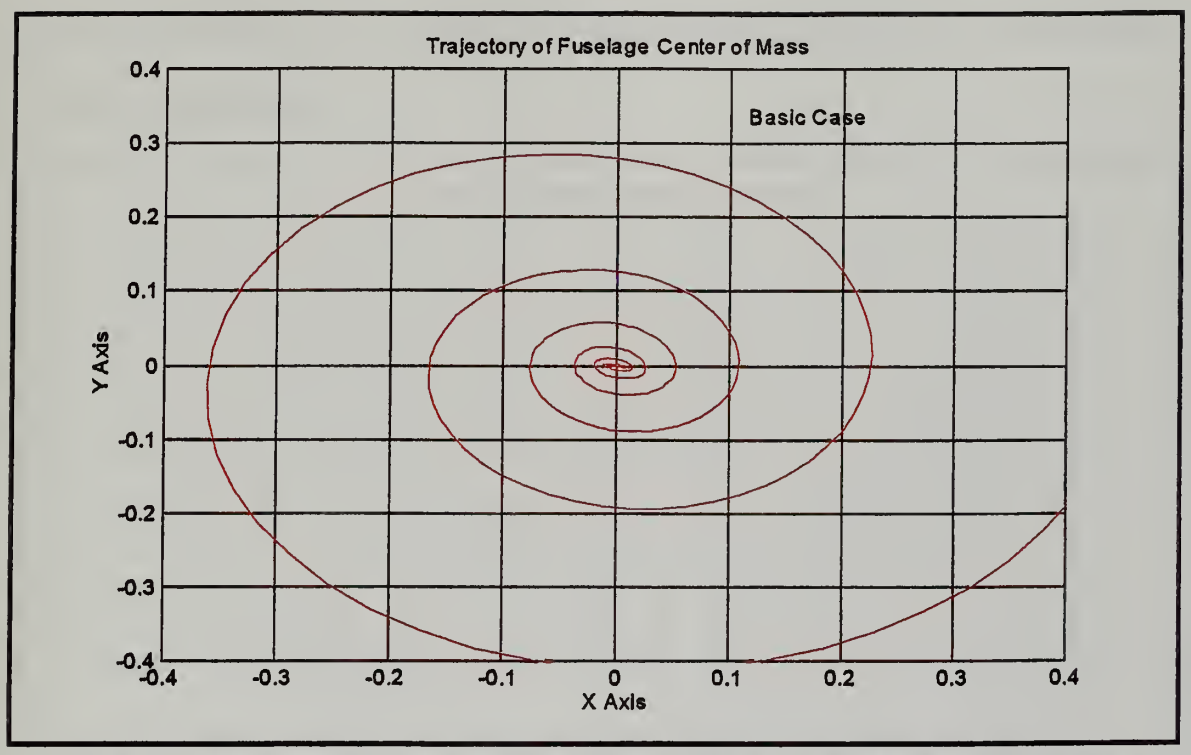


Figure 6.2 Fuselage Trajectory for Basic Parameter Settings, Center of Self Excited Region.

As expected, Figures 6.1 and 6.2 show the rapid divergence of the model as a result of being in the center of the self excited region and perturbed with an initial fuselage velocity in the x-direction. The diverging spiral path of the fuselage center of mass is a characteristic result of the regressing lead-lag mode instability.

Figure 6.3 and 6.4 show the corresponding results for operation below the self excited region. Figures 6.3 shows a beat or modulation of the blade response but no divergence. The beat phenomenon indicates the blade lead-lag motion consists of two dominant modes closely spaced in frequency. The fuselage center of mass trajectory shown in Figure 6.4 shows an elliptical path with the major axis of the ellipse rotating about the zero displacement position. Both the beat phenomenon and the precession type motion of the fuselage seem to be a characteristic behavior of a system operating outside

the self excited region. It is interesting to point out that this behavior is also characteristic of spherical pendulums.

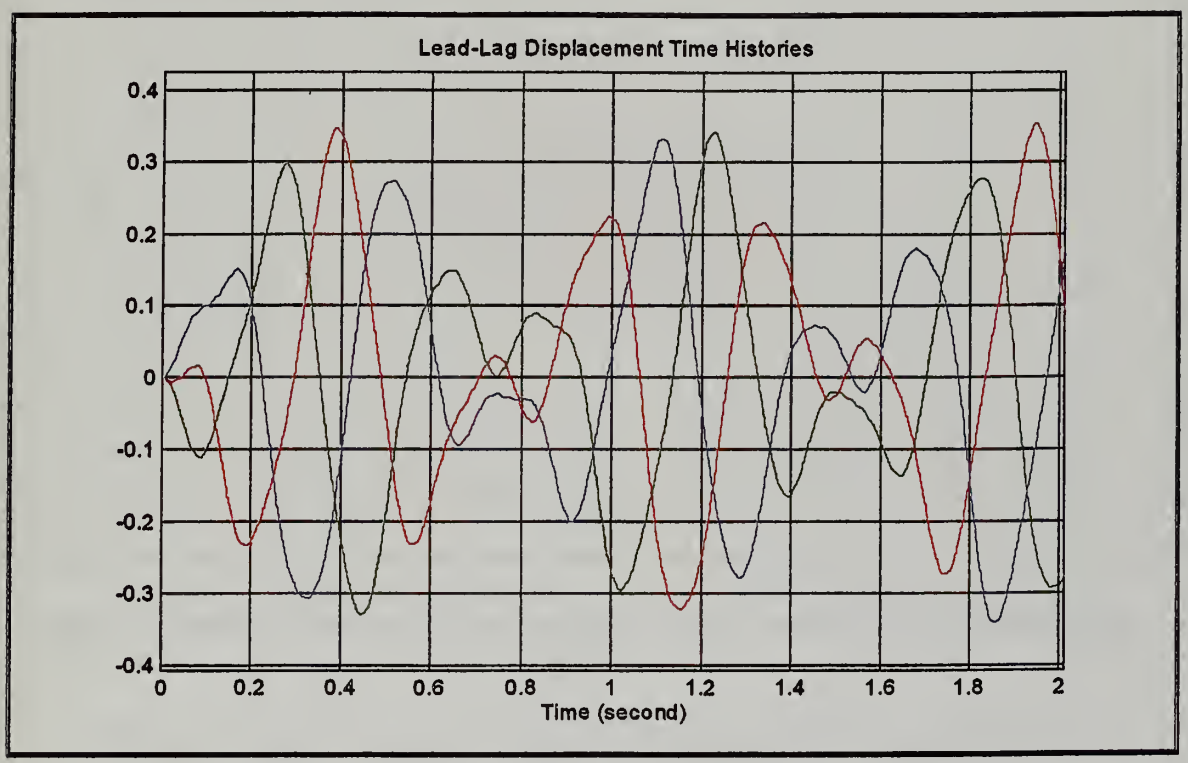


Figure 6.3 Rotor Lead-lag Time Histories, Rotor Speed Below Self Excited Region

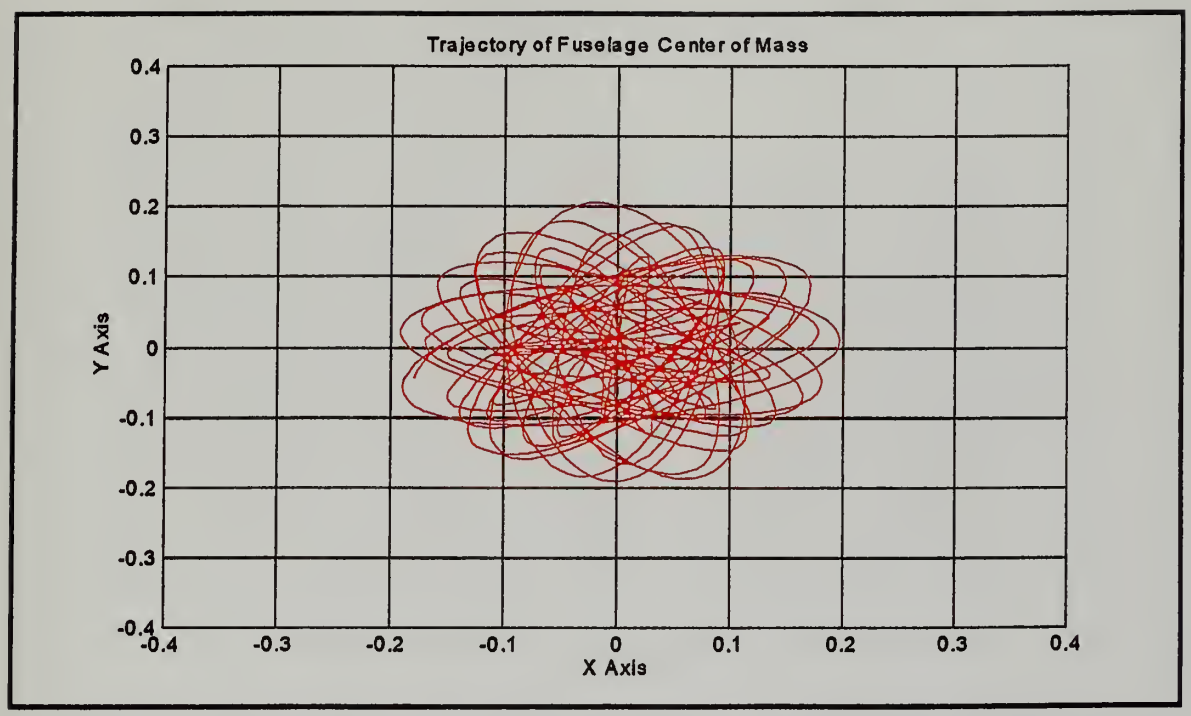


Figure 6.4 Fuselage Trajectory for Basic Parameter Settings, Rotor Speed Below Self Excited Region

Figures 6.5 and 6.6 show the results of a simulation where rotor speed was set above the self excited region. Again, the fuselage exhibits an elliptic whirling motion with the major axis of the ellipse rotating about the zero displacement position while the blade lead lag motion follows a beat pattern.

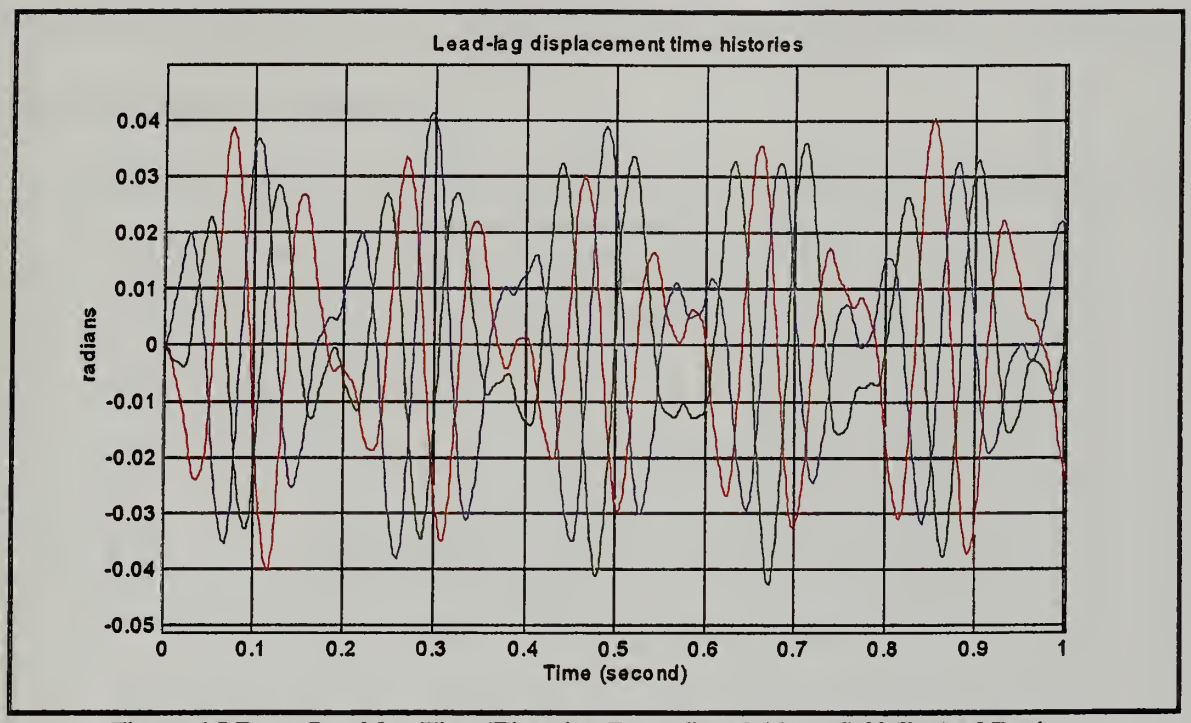


Figure 6.5 Rotor Lead-lag Time Histories, Rotor Speed Above Self Excited Region

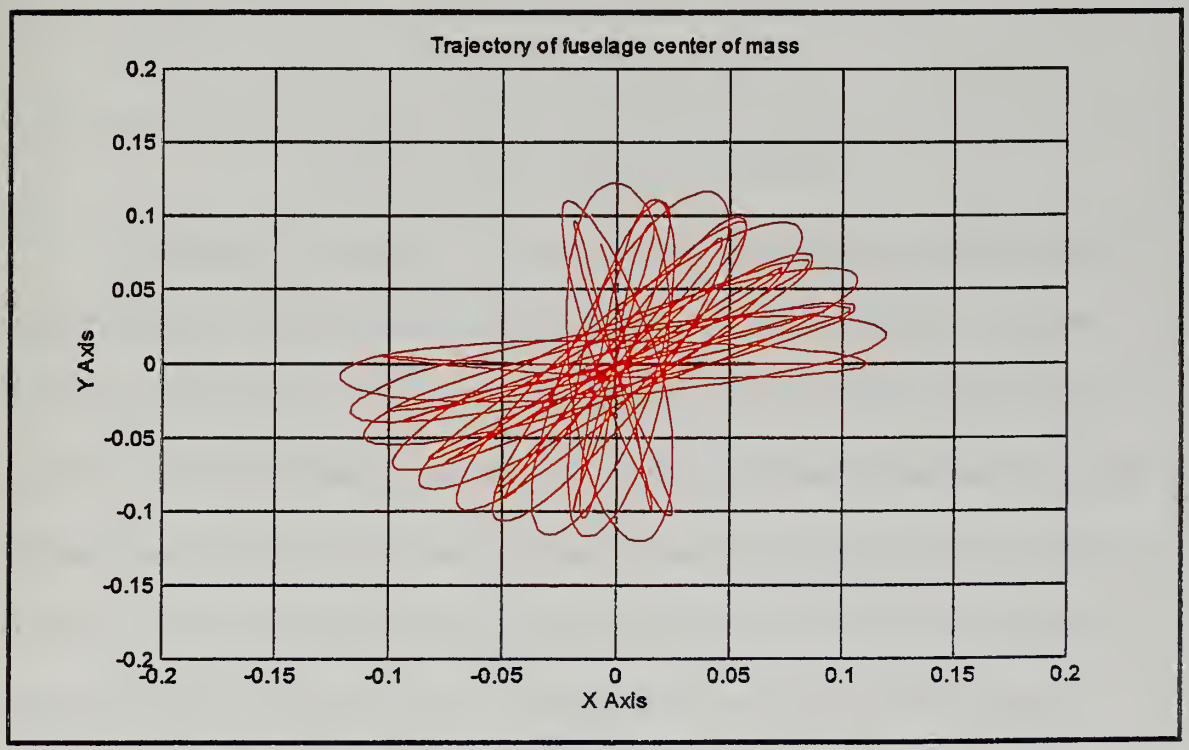


Figure 6.6 Fuselage Center of Mass Trajectory, Rotor Speed Above Self Excited Region

Figure 6.7 is the Coleman stability plot [Ref. 2] for the basic configuration. The red lines indicate the boundaries of the self excited region and the blue line marks the

center of the self excited region. The X's indicate the operating points for the three cases shown in Figures 6.1 through 6.6.

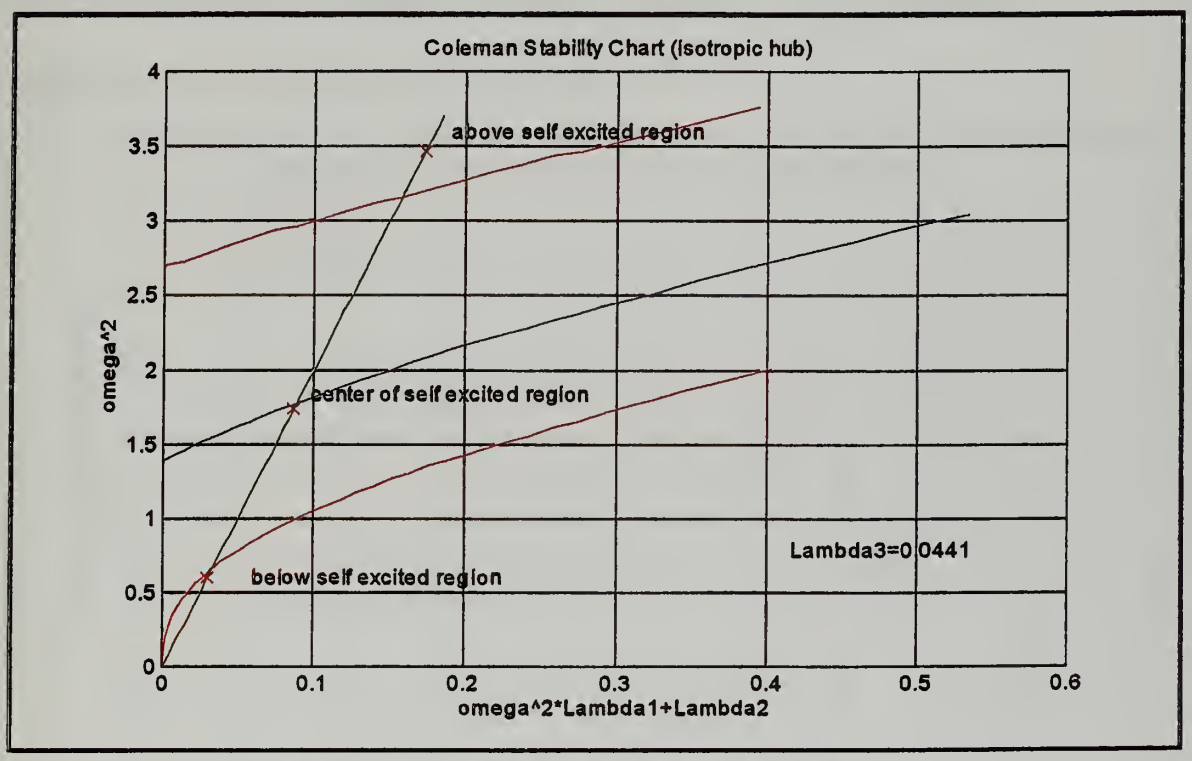


Figure 6.7 Coleman Stability Plot for Basic Case

At this point, a comparison was made between the simulation model and a time history solution of Coleman's and Feingold's equations. Bramwell [Ref. 17] derives Coleman's and Feingold's equation in a form equivalent to that of the simulation model with the blade displacements expressed in the rotating coordinate system and the fuselage displacements expressed in the fixed coordinate system. These equations were solved in the fixed coordinate system using an eigenvalue analysis and the solutions transformed back to rotating coordinate system. A comparison was then made with the lead-lag displacement time history of the simulation model. Figure 6.8 shows the result of the comparison using the parameters of the basic configuration with a moderate amount of damping added to rotor blades and fuselage. Figure 6.8 shows excellent agreement

between the two solutions with a significant departure between the two occurring only when displacements get very large. Thus, for the limiting case of an isotropic hub with linear spring stiffness and damping, the above comparison offers some amount of verification as to the accuracy of the simulation model.

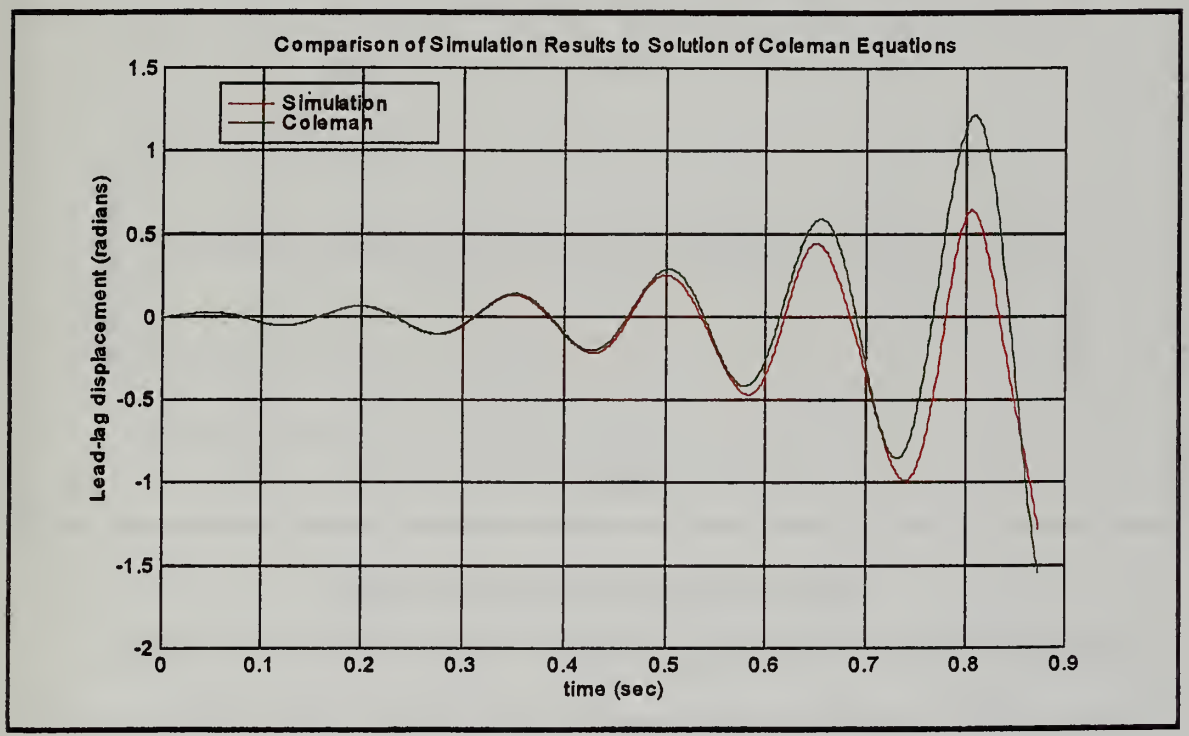


Figure 6.8 Comparison of Simulation Model to Coleman's Model

Moving on from the basic results and model verification, some of the more interesting cases that were simulated will now be discussed. Figure 6.9 shows a comparison between a case where all blade lead-lag dampers are operating and a case where one damper is inoperative. The first plot of Figure 6.9 shows a rotor with all blade dampers operating, in the second plot, the blue damper is disabled by reducing the damping coefficient by two-thirds. As is evident from the plot, the very slightly unstable case with full damper operation is made highly unstable by failing one damper.

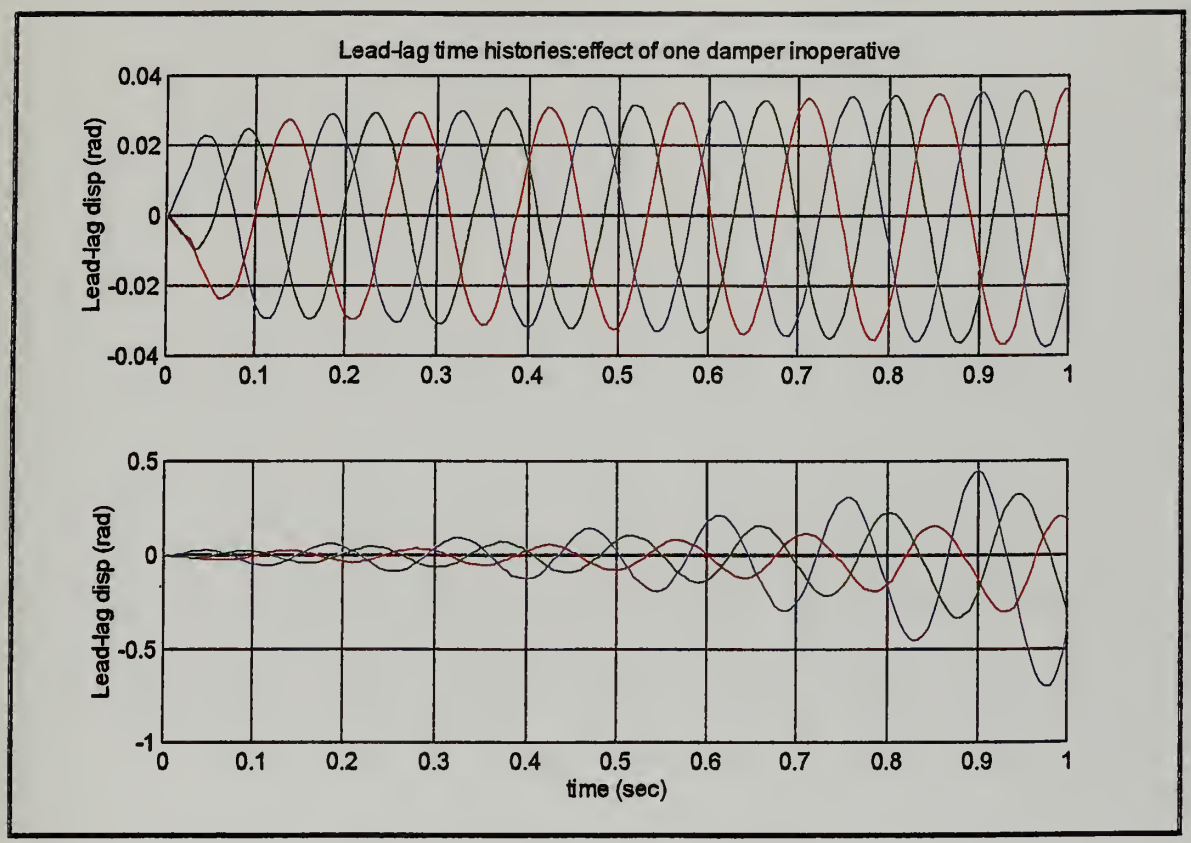


Figure 6.9 One Lead-Lag Damper Inoperative

Figure 6.10 shows the results of simulating damage to a rotor blade by reducing the mass of the blue blade by 20%. The undamaged blades are forced to oscillate around a non-zero displacement position in order to compensate for the damaged blade, but the amplitudes of all the blade oscillations appear to be constrained.

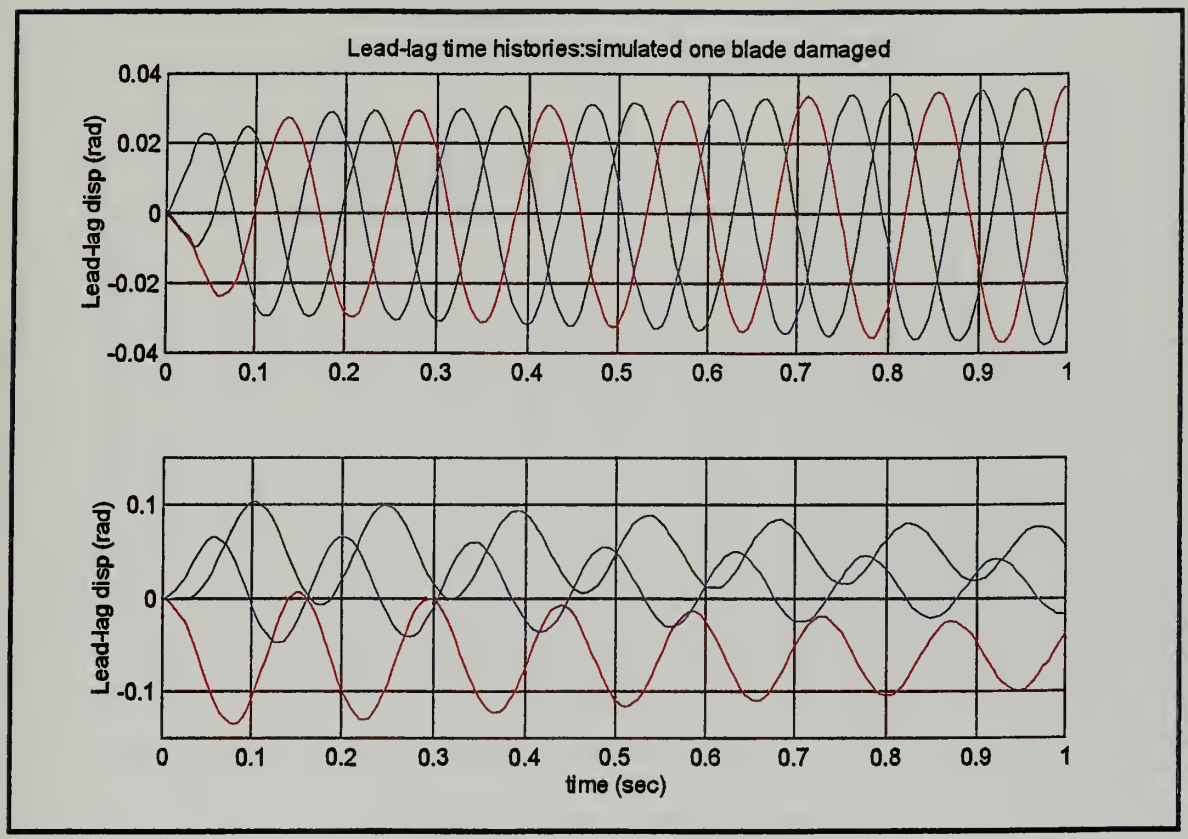


Figure 6.10 Simulated One Rotor Blade Damaged

Figure 6.11 shows the effect of enabling lead-lag stops in the model. The figure compares the time history of a blade with no stops with that of a blade with stops simulated at ±15 degrees. Figure 6.12 shows the corresponding fuselage displacements.

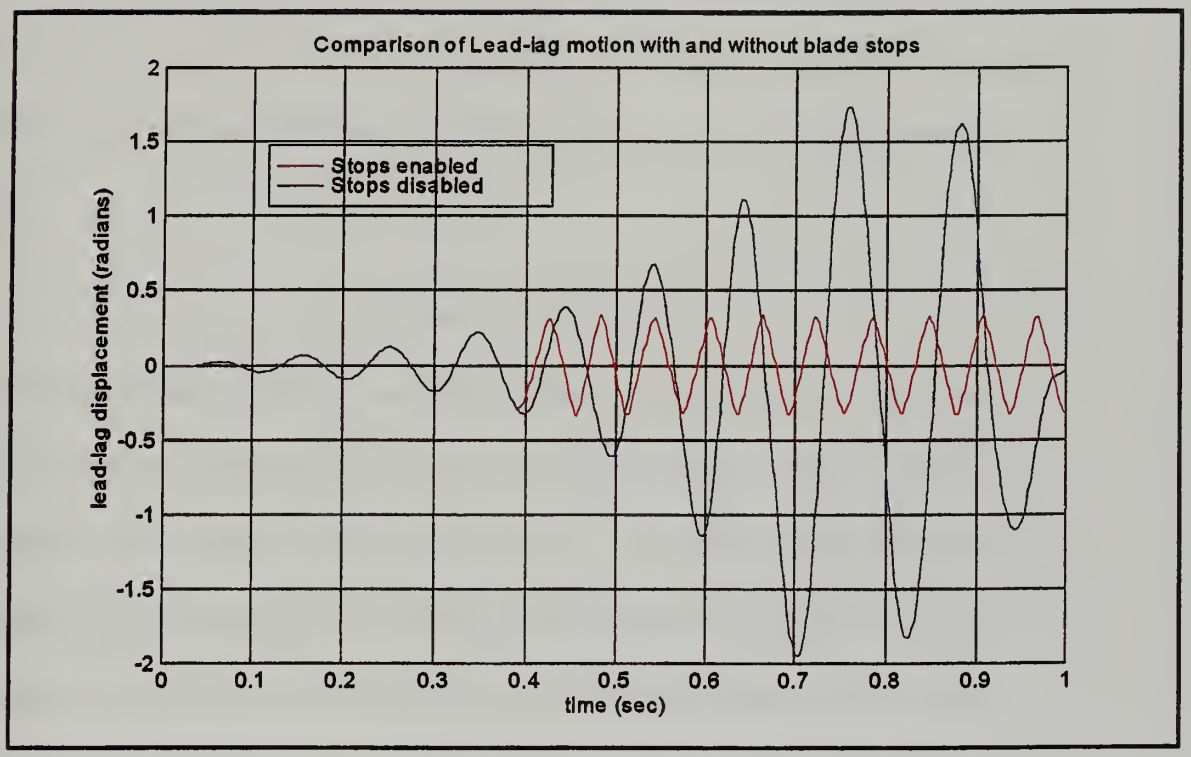


Figure 6.11 Effect of Lead-Lag Stops

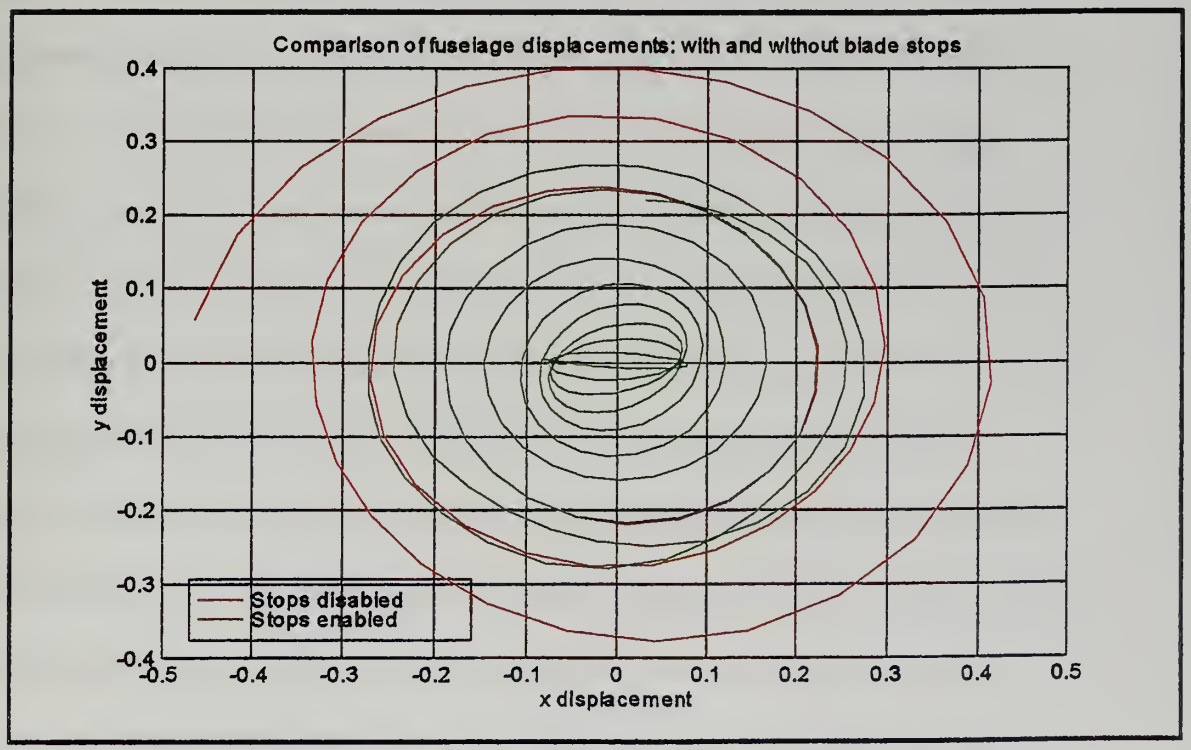


Figure 6.12 Fuselage Displacements with and without Simulated Lead-lag Stops

The objective of the next set of simulations was to examine the effect of a nonlinear flexbeam incorporated into a bearingless rotor design. The nonlinear behavior of the flexbeam was assumed to be that of a Duffing spring where the restoring moment is given by

$$\left(M_\zeta\right)_{Duffing} = K_e\zeta + K_d\zeta^3 \tag{6.1}$$

K_e is the linear stiffness and K_d the nonlinear stiffness. Simulations were conducted for several values of the nonlinear spring constant keeping the linear coefficient constant at 22,000 ft-lbs/radian. Results are shown in Figure 6.13. The primary effect of increasing the nonlinear spring constant is in the limiting of the amplitude of the lead-lag response. As is depicted in the Figure 6.13, the case for $K_d = 0$ is very unstable and a helicopter caught in ground resonance in such a configuration would most likely experience catastrophic failure. By adding the hardening (cubic) term, the unbounded growth in amplitude can be checked as is apparent from the responses for the cases of K_d=4E+5 and K_d=8E+5. As the amplitude increases, the magnitude of the nonlinear term becomes more influential and effectively changes the frequency of oscillation, shifting it outside of the unstable region and allowing the oscillations to decay. Once the amplitude decays to where the influence of the nonlinear term becomes small the cycle repeats itself. While the limiting amplitudes for the nonlinear cases of Figure 6.13 are still large for lead-lag displacements (on the order of 30 to 40 degrees), this limiting behavior may be enough to prevent destruction of an aircraft if ground resonance were to be excited. In flight , when lead-lag displacements are small, the hardening effect of a nonlinear flexbeam would be negligible, and could be designed to act as soft inplane in order minimize hub moments.

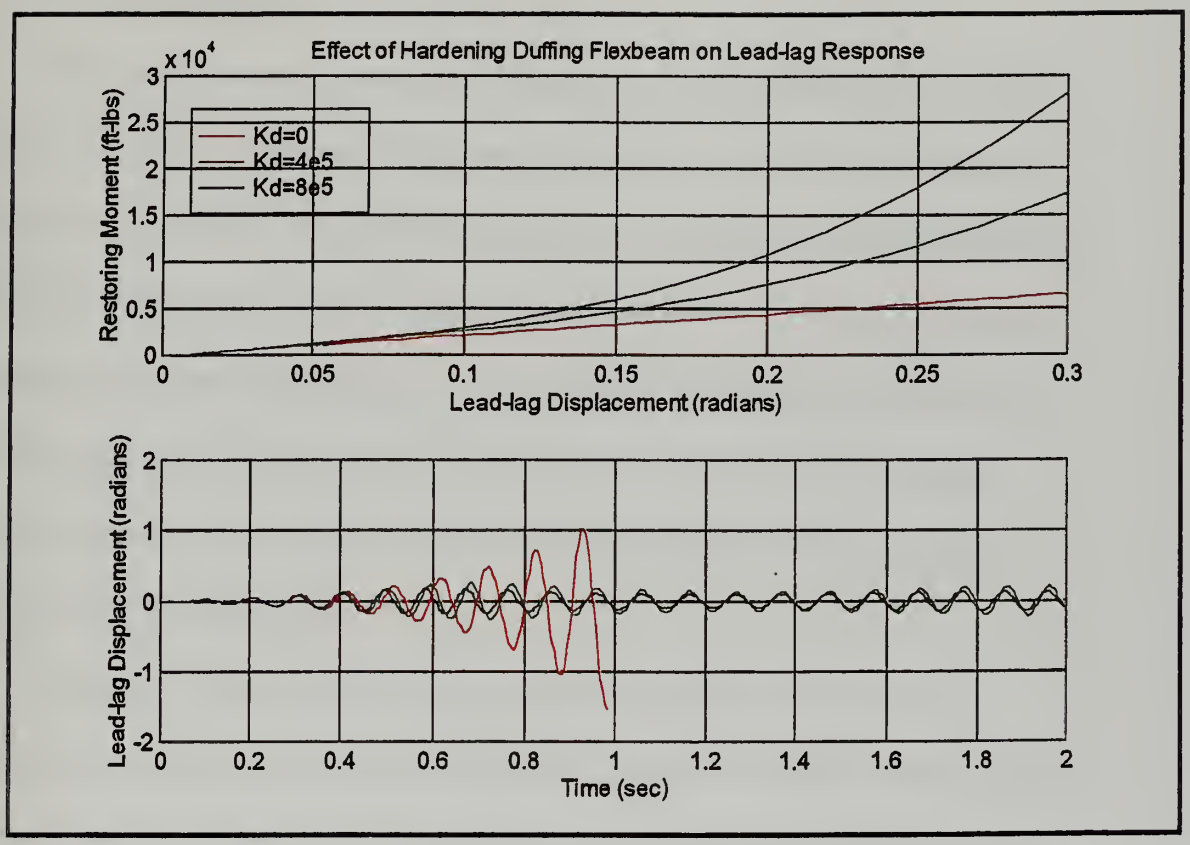


Figure 6.13 Effect of Hardening Duffing Flexbeam on Lead-lag Response

It is important to note that the elastic behaviors of the flexbeams modeled by the curves in the upper plot of Figure 6.13 are purely hypothetical and were selected arbitrarily in order to illustrate the effect that nonlinear elastic behavior could have on rotor system response and stability.

B. COMPLEX MODEL

The complex model is based on a configuration used by Straub [Ref. 12]. The computer code nomenclature and parameter values for the basic case used to conduct simulations for this study is contained in Appendix F. Appendix F is an example of a the MATLAB® input file used for the complex model. This method of input, as opposed to the graphical interface masking feature used for the simple model [Ref.15], was more convenient in the case of the complex model due to the large number of parameters. What follows are examples of some of the time histories generated from simulations completed with the complex model.

Figures 6.14 and 6.15 show the flap and lead-lag response of the rotor to a fuselage roll perturbation (initial angular displacement about the fuselage x-axis). For this case the ground resonance was not excited and both the flap and lead-lag motions settle very quickly. Notice that the lead-lag displacement settles around a non-zero steady state position. This is due to the aerodynamic drag on the rotor blade, the modeling of which is discussed in the next section.

The next set of figures show the results for the same configuration used for Figures 6.14 and 6.15, but the rotor rotational speed has been changed so that ground resonance is excited by the roll perturbation. Figures 6.16 and 6.17 show the flap and lead-lag response for this case, and Figure 6.18 shows the trajectory of fuselage pitch and roll displacements

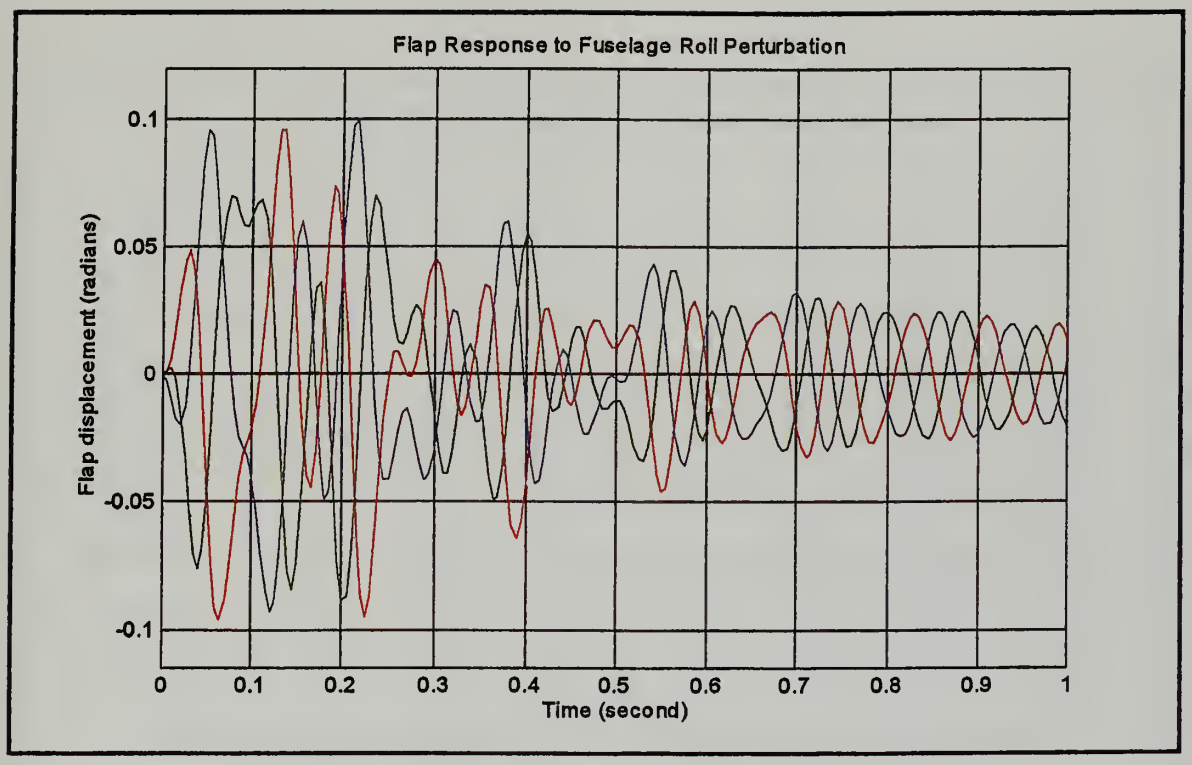


Figure 6.14 Flap Response to Fuselage Roll Perturbation

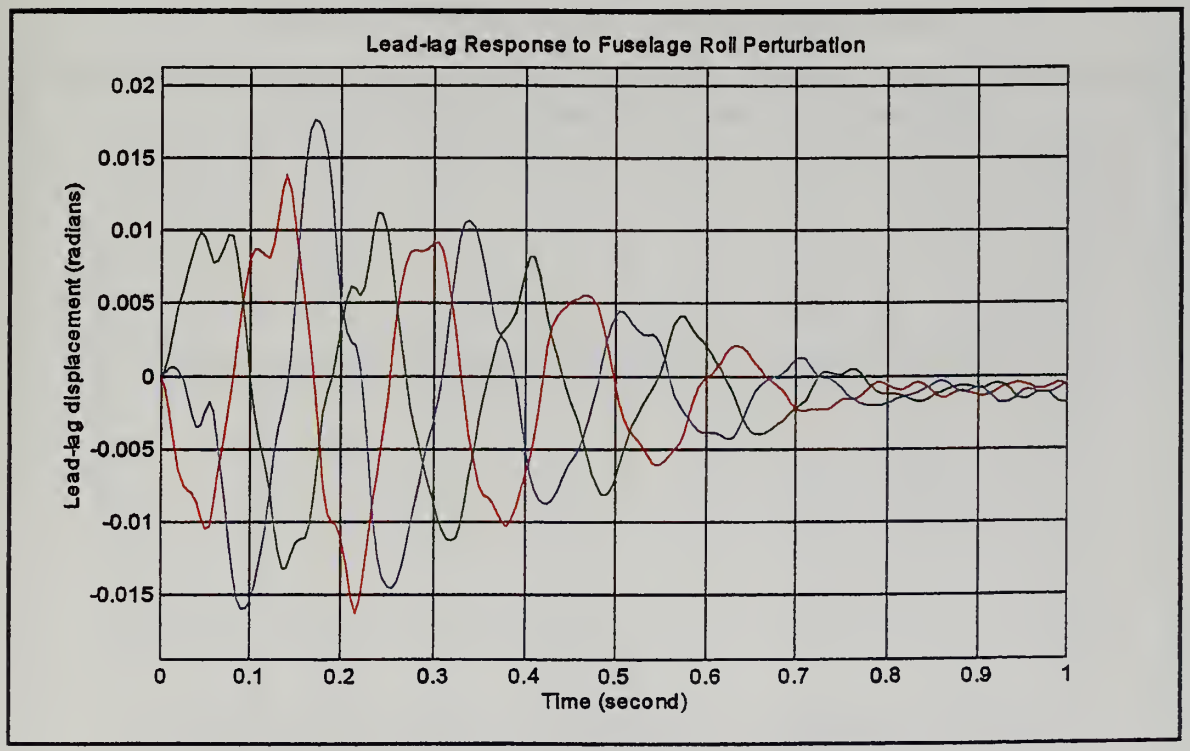


Figure 6.15 Lead-lag Response to Fuselage Roll Perturbation

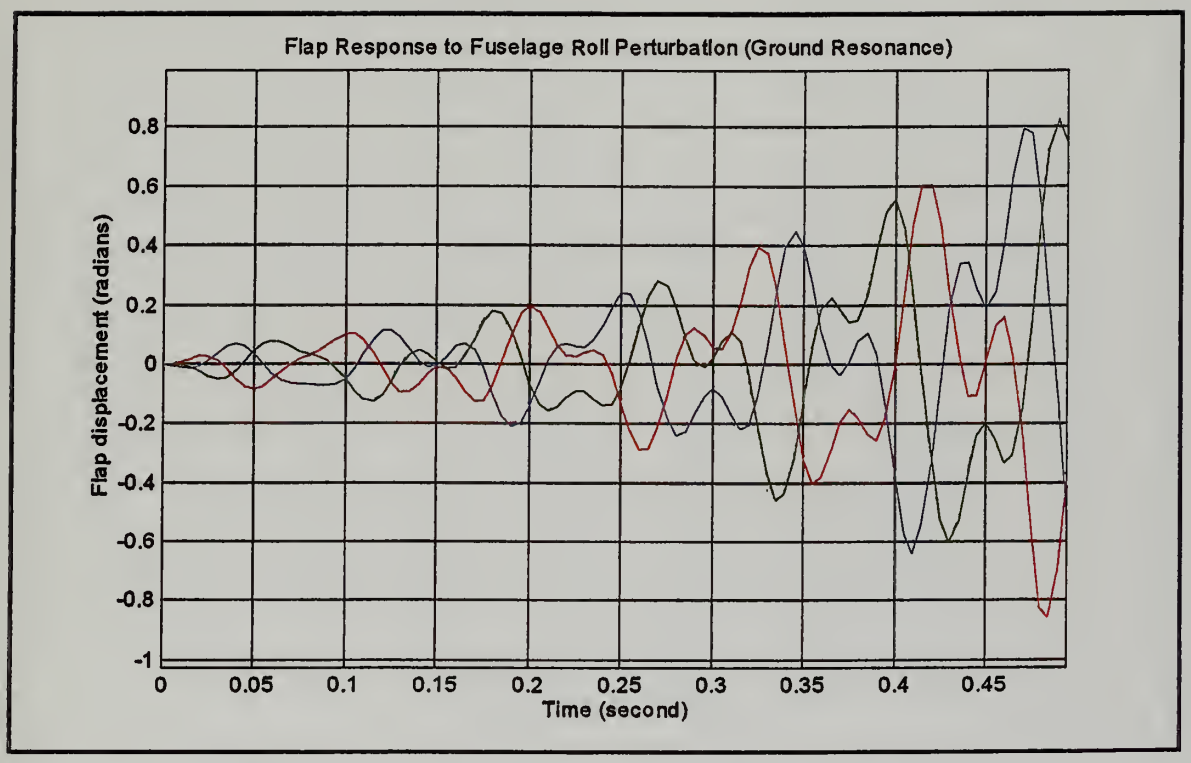


Figure 6.16 Flap Response with Ground Resonance Excited

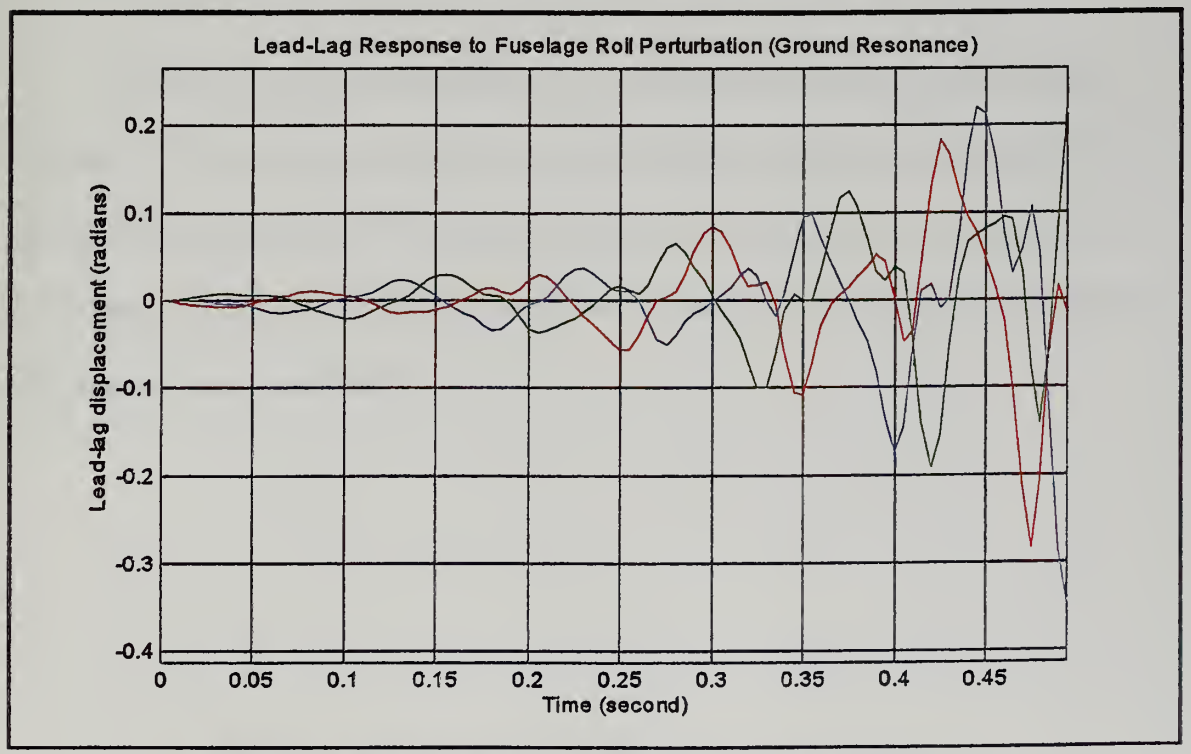


Figure 6.17 Lead-lag Response with Ground Resonance Excited

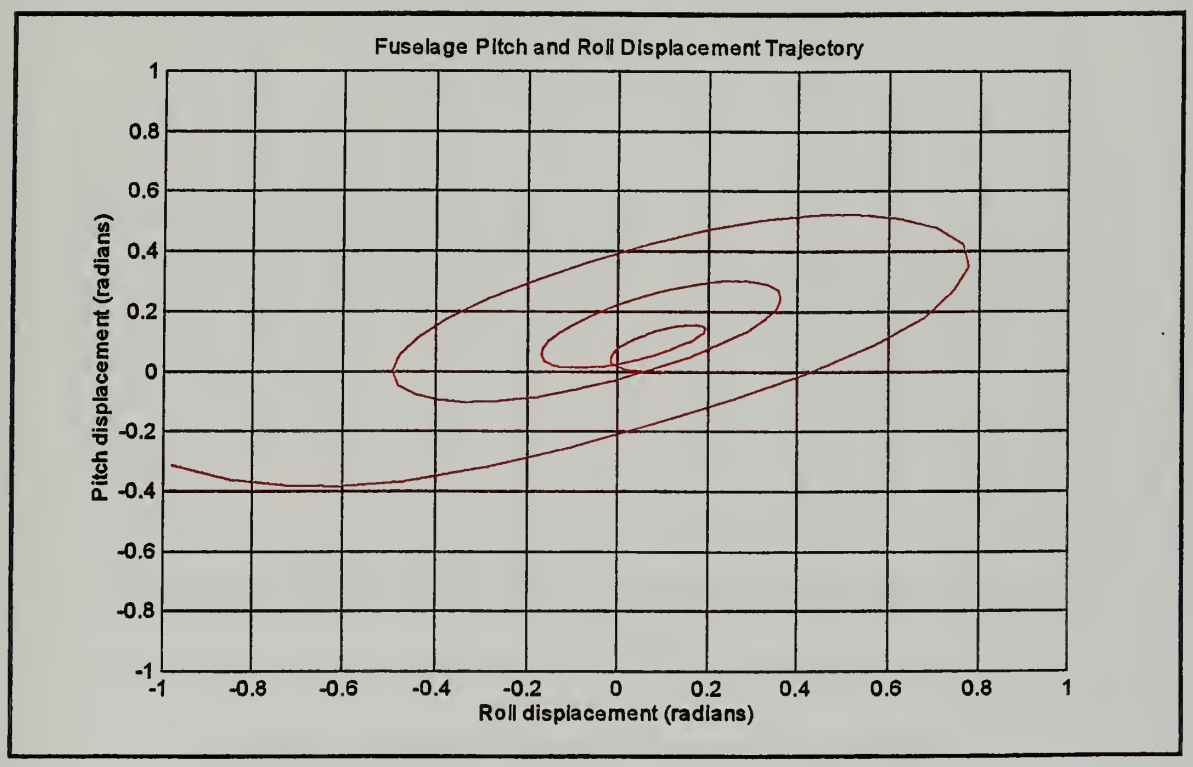


Figure 6.18 Fuselage Displacement Trajectory with Ground Resonance Excited

The next result, shown in Figure 6.19, is the lead-lag response with one damper inoperative. For this case, the system has enough inherent stability to settle out after the initial fuselage displacement. The undamped blade (blue blade) simply settles out at a higher amplitude, but this amplitude is small enough that the inertial forces that arise from the imbalance have a small effect.

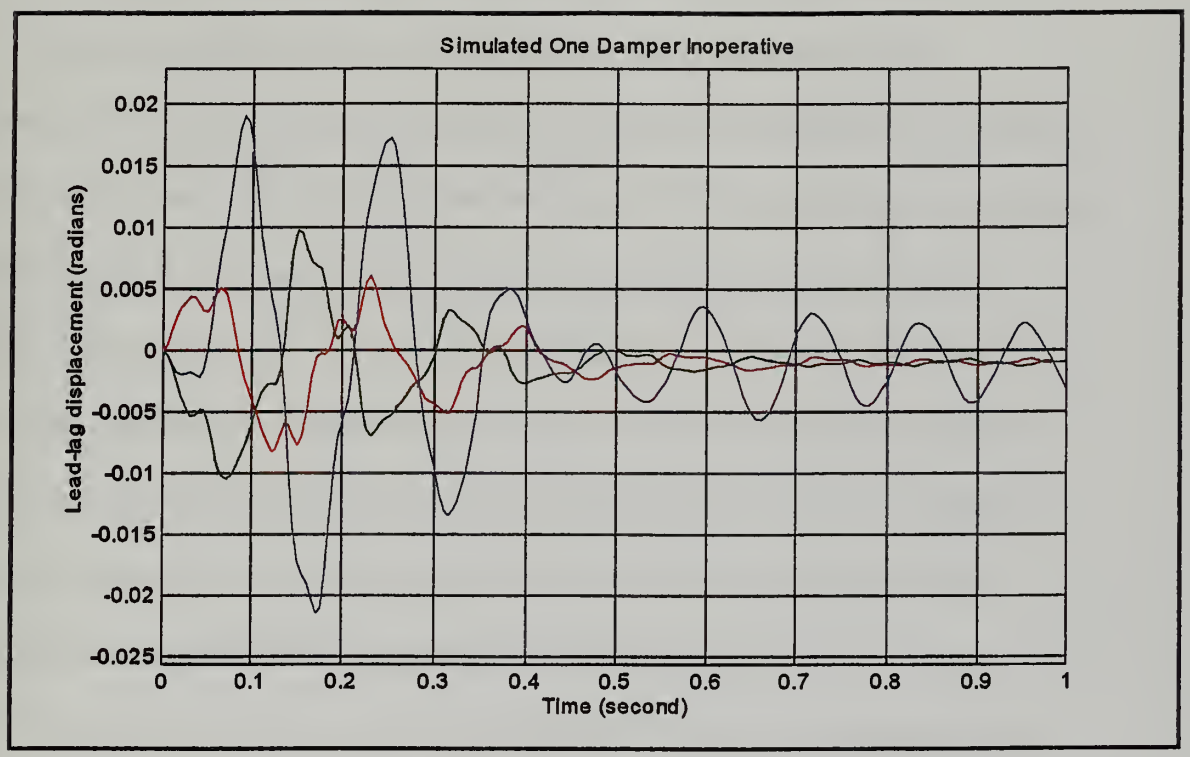


Figure 6.19 Lead-lag Response with One Damper Inoperative

VII. INTRODUCING AERODYNAMICS TO THE MODEL

Aerodynamic forces were derived using quasi-steady strip theory. Stall, compressibility, reversed flow, and wake effects are ignored, and induced flow is obtained from momentum theory. The results of the derivation are the aerodynamic moments about the blade hinge in the flap and lead-lag directions. These moments are entered as generalized aerodynamic forces in the Lagrangian derivation. The development of the aerodynamic equations was adapted from the approach utilized by Kaza and Kvaternik [Ref. 20] for a flap-lag stability analysis on rigid articulated blades. The main difference between their approach and the approach outlined in the following paragraphs is that fuselage motion and its influence on blade motion is considered here.

The relative air velocity at a point on the k^{th} rotor blade due to forward flight at any instant in time, expressed in inertial coordinates is

$$\left(\vec{V}_{air}\right)_I = \mu\,\Omega R e_x + 0 e_y + \lambda\,\Omega R e_z \tag{7.1}$$

The total velocity relative to a blade element is obtained if the above expression is added to the negative of the time derivative of the instantaneous position of the blade element. In inertial coordinates the total velocity is given as

$$\left(\vec{V}_{total}\right)_I = \left(\vec{V}_{air}\right)_I - \left(\dot{\vec{\rho}}\right)_I \tag{7.2}$$

This velocity is expressed in blade deformed coordinates through the following transformation,

$$\left(\vec{V}_{total}\right)_{Bd} = T_3(\zeta_k)T_2(\beta_k)T_3(\psi_k)T_1(r_1)T_2(r_2)\left(\vec{V}_{total}\right)_I \tag{7.3}$$

where T_1, T_2, and T_3 are given by equations (3.1) to (3.3). $\left(\vec{V}_{total}\right)_{Bd}$ can also be expressed as

$$\left(\vec{V}_{total}\right)_{Bd} = U_R\hat{e}_x - U_T\hat{e}_y - U_p\hat{e}_z \tag{7.4}$$

Thus, from the components of the vector expression (7.4), the radial, tangential, and perpendicular components of the total velocity, with respect to blade deformed coordinates, are given.

The lift and drag acting on an elemental section of blade are

$$dL = \frac{1}{2}\rho acU^2\alpha\, dr \tag{7.5}$$

$$dD = \frac{1}{2}\rho acU^2\left(\frac{C_{d_0}}{a}\right)dr \tag{7.6}$$

where

$$U = \sqrt{U_T{}^2 + U_P{}^2} \tag{7.7}$$

The angle of attack, α, is

$$\alpha = \theta - \tan^{-1}\left(\frac{U_P}{U_T}\right) = \theta - \phi \tag{7.8}$$

where θ is the section pitch angle. The lift and drag are then transformed to give the resultant forces along the $\hat{y}$ and $\hat{z}$ axes of the blade deformed coordinate system, thus giving

$$dF_{\hat{y}} = -dL\sin(\phi) - dD\cos(\phi) \tag{7.9}$$

$$dF_{\hat{z}} = dL\cos(\phi) - dD\sin(\phi) \tag{7.10}$$

To obtain the generalized aerodynamic forces from the above expressions, the principle of virtual work is applied for a flap - lag blade displacement sequence. To accomplish this the blade differential forces given by (7.9) and (7.10) are transformed to the blade

undeformed coordinate system since the generalized blade displacements, $\beta_k(t)$ and $\zeta_k(t)$, are expressed in this frame of reference. The transformation is as follows

$$\left(d\vec{F}\right)_{Bu} = \begin{bmatrix} dF_{\hat{x}} \\ dF_{\hat{y}} \\ dF_{\hat{z}} \end{bmatrix} = T_3\left(\zeta_k\right) T_2\left(\beta_k\right) \begin{bmatrix} 0 \\ dF_{\bar{y}} \\ dF_{\bar{z}} \end{bmatrix} \tag{7.11}$$

Applying the principle of virtual work, the generalized aerodynamic forces on the k^{th} rotor blade are

$$\left(M_\beta\right)_k = \int_0^R \left(d\vec{F}\right)_{Bu} \bullet \left(\frac{\partial \vec{\rho}_{Bu\_Bd}}{\partial \beta_k}\right) = \int_0^R r\cos(\zeta) dF_{\bar{z}} \tag{7.12}$$

$$\left(M_\zeta\right)_k = \int_0^R \left(d\vec{F}\right)_{Bu} \bullet \left(\frac{\partial \vec{\rho}_{Bu\_Bd}}{\partial \zeta_k}\right) = \int_0^R r dF_{\bar{y}} \tag{7.13}$$

where $\vec{\rho}_{Bu\_Bd}$ is the position vector of an arbitrary point on the deformed rotor blade elastic axis with respect to the blade undeformed coordinate system.

To simplify the inclusion of aerodynamics into the model, the integral expressions for the generalized aerodynamic forces appearing in equations (7.12) and (7.13) will be evaluated by assuming the mean value of the forces, $dF_{\bar{y}}$ and $dF_{\bar{z}}$, occur at the $r = 0.7R$ radial position, and that this radial position also corresponds to the center of lift and drag on a rotor blade. With these simplifying assumptions, the resulting generalized aerodynamic forces are

$$\left(M_\beta\right)_k = 0.7R^2 \cos\left(\zeta_k\right)\left(dF_{\bar{z}}\right)_{r=0.7R} \tag{7.14}$$

$$\left(M_\zeta\right)_k = 0.7R^2\left(dF_{\bar{y}}\right)_{r=0.7R} \tag{7.15}$$

The inflow ratio, λ, and advance ratio, μ appearing in equation (7.1) are

$$\lambda = \frac{\left(V \sin\alpha - v_i\right)}{\Omega R} \tag{7.16}$$

$$\mu = \frac{V \cos\alpha}{\Omega R} \tag{7.17}$$

The induced flow, v_i, in equation (7.16) is calculated by equating the integrated thrust to the thrust from momentum theory, leading to the result,

$$v_i = \frac{C_T \Omega R}{2\sqrt{\mu^2 + \lambda^2}} \tag{7.18}$$

The thrust coefficient, C_T, is determined by adding the average total lift generated by each rotor blade over one rotor revolution and dividing the quantity, $\rho \pi \Omega^2 R^4$.

For this study, only a rotor-fuselage system in the ground resonance regime is considered, so the thrust coefficient and forward velocity were set to zero, giving an inflow ratio of zero, which corresponds to a steady state rotor blade pitch angle of $(\theta_k)_{ss} = 0$, assuming an uncambered blade. This simplifies things greatly for this study by eliminating the requirement to trim the rotor system for a certain aircraft weight and flight condition. The aerodynamics came into play in analyzing the effects active pitch inputs about a flat pitch condition on ground resonance stability. This will be discussed further in a later section.

VIII. MOVING BLOCK TECHNIQUE

One of the drawbacks of performing direct numerical simulation of dynamic systems is that time histories of system degrees of freedom only offer qualitative information on the effect that certain system parameters have on system stability or performance. In order to quantify the effects of varying certain system parameters, such as rotor speed, flex-beam stiffness, and active control inputs, on rotor-fuselage stability in the ground resonance regime, a method was needed to estimate system damping levels from the system time histories. Moving Block Analysis, a technique developed at Lockheed in the 1970's, is a digital method of analyzing a transient time history to obtain modal damping and frequency. The technique is first described in some detail by Hammond and Dogget [Ref. 18].

The technique is analytically based on the typical transient response of a second order system. Consider the following transient time history,

$$y(t) = A\, e^{-\varsigma\omega_n t} \sin(\omega t + \vartheta) \tag{8.1}$$

where

$$\omega^2 = \omega_n{}^2\left(1-\varsigma^2\right) \tag{8.2}$$

The finite Fourier transform of $y(t)$ at the damped frequency, ω, from time τ to $\tau + T$ is,

$$F(\omega) = \int_{\tau}^{\tau+T} A e^{-\varsigma\omega_n t} \sin(\omega t + \vartheta)\, e^{-i\omega t} dt \tag{8.3}$$

After carrying out the integration and making the assumptions that $\zeta << 1$, so $\omega_n \approx \omega$, the magnitude of the Fourier transform can be written in the following form,

$$|F(\omega)| = \frac{A}{2\omega} e^{-\varsigma\omega\tau} \left[\frac{1+f(\varsigma)}{\varsigma^2} \right]^{\frac{1}{2}} \tag{8.4}$$

where

$$f(\varsigma) = -2e^{-\varsigma\omega T} + e^{-2\varsigma\omega T} + \left(1-e^{-\varsigma\omega T}\right)\varsigma \sin\left(2\left(\omega\tau+\vartheta\right)\right) - e^{-\varsigma\omega T}\left(1-e^{-\varsigma\omega T}\right)\varsigma \sin\left[2\left(\omega(\tau+T)+\vartheta\right)\right] \tag{8.5}$$

Taking the natural logarithm of equation (8.4) yields

$$\ln|F(\omega)| = -\varsigma\omega\tau + \ln\left(\frac{A}{2\omega}\right) + \frac{1}{2}\ln\left(\frac{1+f(\varsigma)}{\varsigma^2}\right) \tag{8.6}$$

The last term in equation (8.6) can be expanded in a Taylor series to yield

$$\begin{aligned} \ln|F(\omega)| &= -\varsigma\omega\tau + \ln\left(\frac{A}{2\omega}\right) \\ &+ \frac{1}{2}\ln\left[(\omega T)^2 + (\omega T)\left\{\sin\left(2\left(\omega\tau+\vartheta\right)\right) - \sin\left(2\left[\omega(\tau+T)+\vartheta\right]\right)\right\}\right] \\ &- \frac{\varsigma}{4}\omega T\left[\frac{2\omega T + \sin\left(2\left(\omega\tau+\vartheta\right)\right) - 3\sin\left[2\left(\omega(\tau+T)+\vartheta\right)\right]}{\omega T + \sin\left(2\left(\omega\tau+\vartheta\right)\right) - \sin\left[2\left(\omega(\tau+T)+\vartheta\right)\right]}\right] \end{aligned} \tag{8.7}$$

From equation (8.7) it can be seen that if $\ln|F(\omega)|$ is plotted versus τ, the resulting curve will be the superposition of a straight line with slope $-\varsigma\omega$ and an oscillatory component which oscillates about the straight line with frequency 2ω. If it is assumed that T is an integral multiple of the basic period of oscillation, such that

$$T = \frac{2\pi N}{\omega} \tag{8.8}$$

equation (8.7) reduces to

$$\ln|F(\omega)| = -\varsigma\omega\tau + \frac{1}{2}\varsigma \sin\left(2\left(\omega\tau+\vartheta\right)+C\right) \tag{8.9}$$

where C is a constant given by

$$C = \ln\left(\frac{A}{2\omega}\right) + \ln(\omega T) - \frac{\varsigma\omega T}{2} \tag{8.10}$$

From equations (8.9) and (8.10) it can be seen that if successive discrete Fourier transforms at a frequency ω are performed for $0 \le \tau \le t_s - T$, where t_s is the total signal length, a plot can be made from which the damping can be determined. It is this procedure which is the basis for the moving block analysis [Ref. 18].

For a sampled signal, the moving block method is applied by first estimating the frequency of interest embedded in the signal using a Fast Fourier Transform (FFT). A block length is selected consisting of N_b data points, and the moving block function, $\ln|F(\omega,\tau)|$, is calculated for $\tau = 0$. The block is then shifted one data point (time step) at a time and the moving block function recomputed for $\tau = n\Delta t$, where n = 0, 1, 2, ..., N - N_b. The plot of $\ln|F(\omega,\tau)|$ versus τ is fitted with a linear least squares fit, and the damping is estimated from the slope of the curve [Ref. 19].

For this study, a moving block analysis program was developed with MATLAB®. The m-file code is contained in Appendix E. Moving block was applied in the code by adapting and combining the procedures outlined by Hammond and Doggett [Ref.18] and Bousman and Winkler [Ref. 19]. The method used for computing the moving block function once the frequency and block size were determined was to evaluate the Fourier coefficients for the first time block with the following relations

$$a_k(\tau) = \frac{2}{N_b}\sum_{x=0}^{N_b-1} f(x+\tau)\cos\frac{2\pi k_b x}{N_b} \tag{8.11}$$

$$b_k(\tau) = \frac{2}{N_b} \sum_{x=0}^{N_b-1} f(x+\tau) \sin \frac{2\pi k_b x}{N_b} \tag{8.12}$$

where

$$k_b = \omega N_b \Delta t \tag{8.13}$$

and $f(x+\tau)$ is the signal data (for the first block $\tau = 0$). The Fourier coefficients at the next time step are then calculated using the following recursion relations,

$$a_k(\tau+1) = \left\{ a_k(\tau) + \frac{2}{N_b} \left[f(N_b+\tau) - f(\tau) \right] \right\} \cos\left(\frac{2\pi k_b}{N_b} \right) + b_k(\tau) \sin\left(\frac{2\pi k_b}{N_b} \right) \tag{8.14}$$

$$b_k(\tau+1) = -\left\{ a_k(\tau) + \frac{2}{N_b} \left[f(N_b+\tau) - f(\tau) \right] \right\} \sin\left(\frac{2\pi k_b}{N_b} \right) + b_k(\tau) \cos\left(\frac{2\pi k_b}{N_b} \right) \tag{8.15}$$

The magnitude of the natural logarithm of the moving block function is then given by

$$\ln\left[F(\omega,\tau) \right] = \frac{1}{2} \ln\left[a_k(\tau)^2 + b_k(\tau)^2 \right] \tag{8.16}$$

The accuracy and speed of the moving block analysis code is dependent on several factors. The frequency resolution of the FFT algorithm is inversely proportional to the signal length and sampling rate, or,

$$\Delta\omega = \frac{1}{N\Delta t} \tag{8.17}$$

For the current study, signal frequencies will range from approximately 5 to 30 Hz, and sampling rates will be between 100 and 2000 Hz with the typical record lengths of 2 to 5 seconds. The worst resolution considering these figures would be approximately 0.5 Hz, an error of 10% for the low frequency signal. To compensate for this, a refinement procedure [Ref. 18] is incorporated into the code. Additionally, the FFT algorithm is optimized to operate on record lengths in powers of two. If the record length is not a

power of two, it is automatically padded with the required number of zeros, which degrades the accuracy of the frequency estimation. To remedy this, all signal lengths were controlled to be exactly in powers of two by adjusting stop time and the size of time steps when executing simulations.

The moving block code developed for this study was specialized for handling uni-modal or bi-modal signals, i.e., signals with one to two dominant modes. For the bi-modal case, sufficient frequency separation must exist such that the resolution offered by the methods described above will be adequate enough for accurate damping estimates of both modes. A test signal of the following form was used to verify the accuracy of program,

$$f(t) = A_1 \exp\left(\frac{-2\pi\omega_1\varsigma_1 t}{\sqrt{1-\varsigma_1}}\right)\sin(2\pi\omega_1 t) + A_2 \exp\left(\frac{-2\pi\omega_2\varsigma_2 t}{\sqrt{1-\varsigma_2}}\right)\sin(2\pi\omega_2 t) \qquad (8.18)$$

Parameter values were fixed to match the test case considered in Ref. 9, and were: $A_1 = A_2 = 1000$, $\varsigma_1 = 0.01$, $\varsigma_2 = 0.02$, $\omega_1 = 8.0$ Hz, and $\omega_2 = 6.0$ Hz. The results of the moving block analysis on the test signal shown in Figure 8.1 is shown in Figure 8.2. The first plot of Figure 8.2 shows the resulting power spectrum generated by a FFT of the test signal. The second plot is the result of refining the FFT frequency estimates. The third plot is the crux of the moving block analysis where the negative of the slopes of the straight line least square fits give the damping ratios when divided by the corresponding damped frequencies of the particular mode. The damping and frequencies obtained from Figure 8.2 are summarized in the Table 8.1.

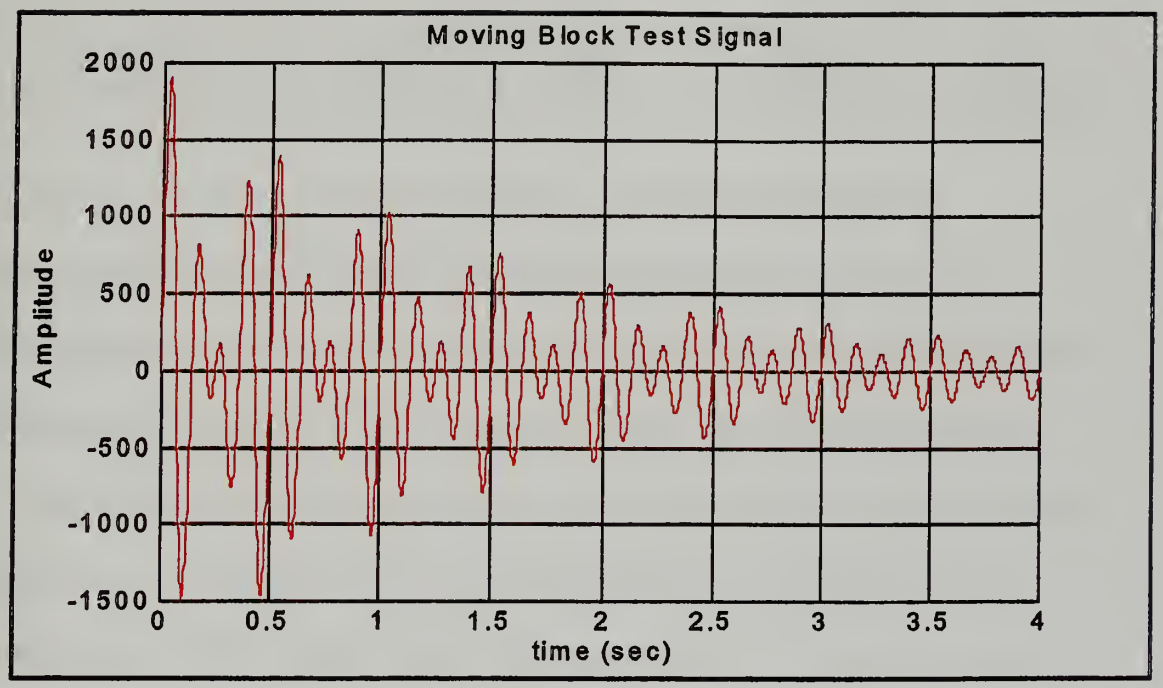


Figure 8.1 Moving Block Test Signal

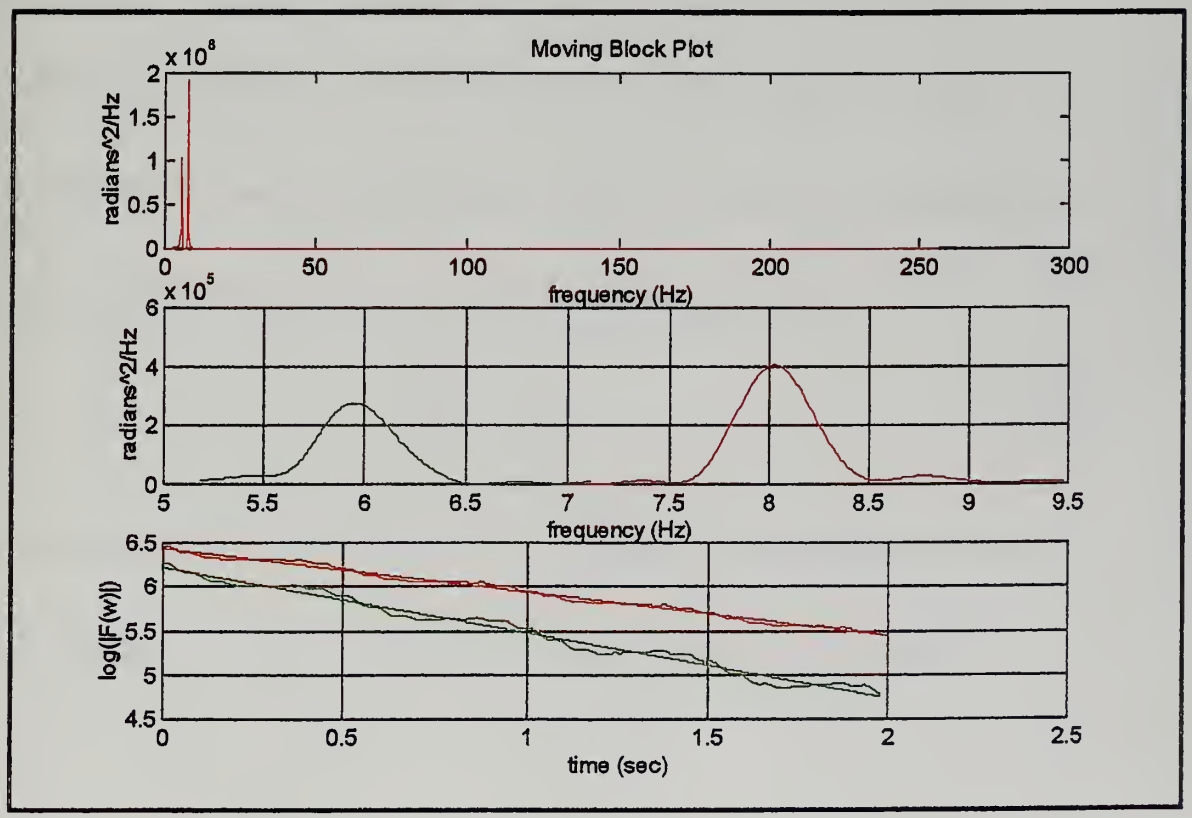


Figure 8.2 Result of Moving Block on Test Signal

Table 8.1 Summary of Results for Moving Block Bi-modal Signal Test Case

Parameter →	ω_1	ς_1	ω_2	ς_2
Test signal	**8 Hz**	**0.01**	**6 Hz**	**0.02**
Moving Block Analysis	**8.0314 Hz**	**0.0098**	**5.9535 Hz**	**0.0198**

From the results, it can be seen that the two modes of the test signal are sufficiently far apart in order to obtain reasonably accurate damping estimates. It is, however, important to note that as the frequency separation of a bi-modal signal decreases to approximately 5% [Ref. 19], accurate damping estimates will no longer be possible.

Figure 8.3 shows the effect of varying rotor speed on first lead-lag mode damping as determined by a moving block analysis. The configuration used for Figure 8.3 is the basic configuration with a moderate amount of damping added to the fuselage and blade motions in order keep time histories within reasonable bounds when simulating inside the self-excited region. The initial excitation was provided by setting an initial fuselage displacement in the x-direction. The rotor frequency (abscissa in Figure 8.3) is non-dimensionalized by the fuselage natural frequency ($\sqrt{\frac{K_x}{M_x}}$). The center of the regressing lead-lag mode instability should occur at a non-dimensional frequency

$$\frac{\Omega}{\Omega_0} = \frac{1}{1-\sqrt{\frac{e}{R}}} \tag{8.19}$$

for a rotor system modeled with point masses. For the basic configuration, the center of instability corresponds to $\frac{\Omega}{\Omega_0} = 1.288$, which is in good agreement with Figure 8.3.

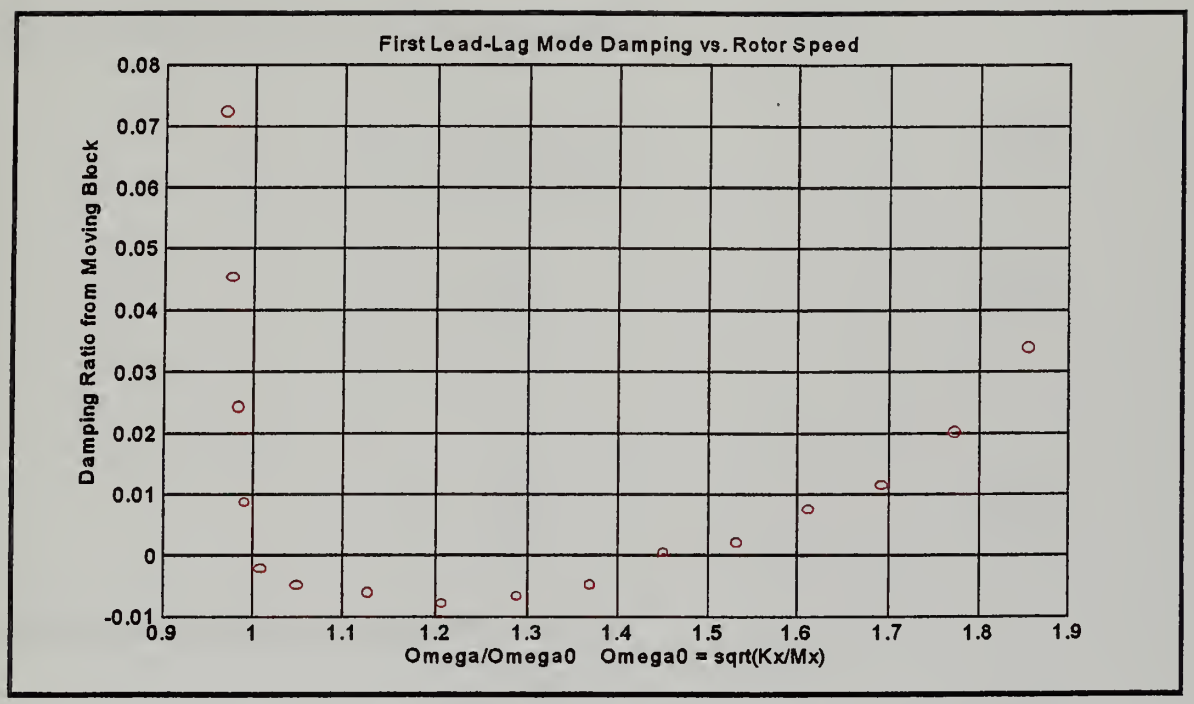


Figure 8.3 Effect of Varying Rotor Speed on First Lead-lag Mode Damping

Figure 8.4 shows the results of a moving block analysis completed for rotor systems with different Deutsch numbers. Deutsch [Ref. 21] derives a criteria to determine the quantity of damping necessary to eliminate ground resonance instability through the full range of rotor speeds. The criteria requires that the product of the blade and fuselage damping parameters be greater than a certain parameter determined by the rotor-fuselage configuration. The rotor-fuselage configuration parameters for the case of the simple model with an isotropic pylon and rotor are given by the following,

$$\Lambda_1 = \frac{e}{R} \tag{8.20}$$

$$\Lambda_2 = \frac{K_\zeta}{I} \tag{8.21}$$

$$\Lambda_3 = \frac{Nm_b}{2\left(M_p + Nm_b\right)} \tag{8.22}$$

$$I = m_b R^2 \tag{8.23}$$

The damping parameters for the pylon and rotor blades are

$$\lambda_p = \frac{C_p}{\left(M_p + N m_b\right)\omega_p} \tag{8.24}$$

$$\lambda_\zeta = \frac{C_\zeta}{I\omega_p} \tag{8.25}$$

where

$$\omega_p = \sqrt{\frac{K_p}{M_p}} \tag{8.26}$$

Deutsch's criteria for elimination of ground resonance is

$$D = \frac{\lambda_p \lambda_\zeta}{\Lambda_3 \Big/ \left(p-1\right)} > 1 \tag{8.27}$$

where

$$p = \frac{1 + \sqrt{\Lambda_1 + \Lambda_2 - \Lambda_1 \Lambda_2}}{1 - \Lambda_1} \tag{8.28}$$

Here, the ratio of the damping product to the configuration parameter is defined as the Deutsch number , D .

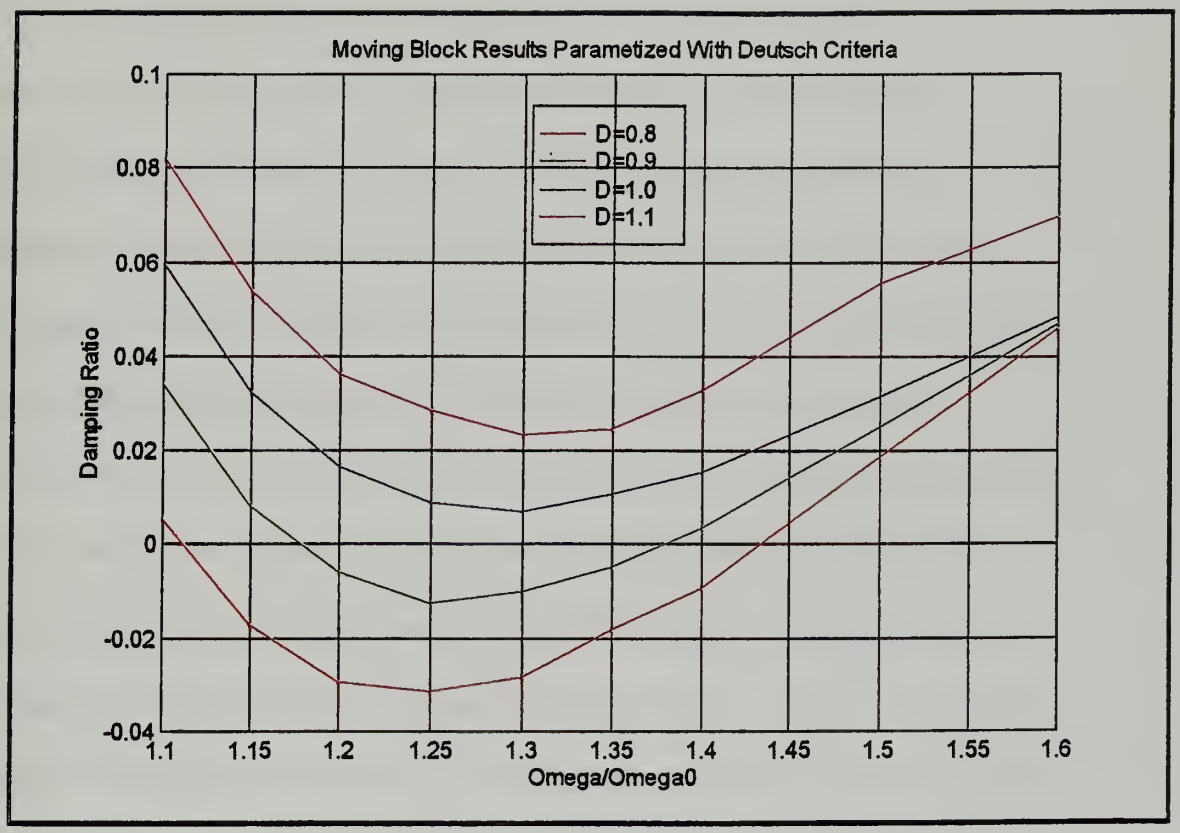


Figure 8.4 Moving Block Results Parametized by Deutsch Number

IX. ACTIVE CONTROL

Several studies have addressed the problem of eliminating helicopter aeromechanical instabilities with active control. Straub [Ref. 12] and Straub and Warmbrodt [Ref. 13] use linear state space methods to investigate the effects of systematically varying feedback gain and phase in a closed loop system on the rotor-fuselage dynamic behavior. Takahashi and Friedmann [Ref. 22] move one step further by proposing a simple closed loop controller based on an optimal state estimator in conjunction with optimal state feedback determined from linear quadratic Gaussian (LQG) optimization techniques. Weller [Ref. 23] showed by experiment that a fixed gain controller which transforms fuselage states into swashplate inputs can greatly improve aeromechanical stability margins and eliminate unstable envelopes. Wood , et al., [Ref. 24] detailed the design and implementation of a higher harmonic pitch control system and demonstrated that it can be an effective means of vibration reduction. For the scope of this study a similar approach to the one used by Weller [Ref. 23] in his experimental investigation was incorporated into the simulation environment.

In the most general case, the complex model allows rotor blade pitch inputs to be independent of one another so that the simulation of individual blade control is possible. The general form of a pitch input is,

$$\theta_k = (\theta_s)_k \sin(n\Omega t + \Phi_k) + (\theta_c)_k \cos(n\Omega t + \Phi_k) \tag{9.1}$$

where n is the harmonic number of the pitch frequency, $(\theta_s)_k$ and $(\theta_c)_k$ are the input phase and amplitude weighting parameters (for a first harmonic input they would correspond to longitudinal and lateral cyclic for the k^{th} rotor blade), and Φ_k is the

azimuthal phase angle of the k^{th} rotor blade. For the following active control example, active pitch inputs via a swashplate were simulated for the a three bladed rotor, thus, n was set equal to unity and the amplitude and phase weighting parameters for each blade were set equal to each other such that,

$$(\theta_s)_1 = (\theta_s)_2 = (\theta_s)_3 = \theta_s \tag{9.2}$$

$$(\theta_c)_1 = (\theta_c)_2 = (\theta_c)_3 = \theta_c \tag{9.3}$$

Pylon pitch and roll position feedback was transformed into swashplate control inputs by the following fixed gain relationship

$$\begin{bmatrix} \theta_c \\ \theta_s \end{bmatrix} = \begin{bmatrix} K\cos(\phi) & K\sin(\phi) \\ -K\sin(\phi) & K\cos(\phi) \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \tag{9.4}$$

Stability measurements were made based on time histories of the orthogonal components of the rotor center of gravity offset given by

$$x_{cg} = \frac{\sum_{k=1}^{N} (m_b)_k (x_{cg})_k}{\sum_{k=1}^{N} (m_b)_k} \tag{9.5}$$

$$y_{cg} = \frac{\sum_{k=1}^{N} (m_b)_k (y_{cg})_k}{\sum_{k=1}^{N} (m_b)_k} \tag{9.6}$$

where,

$$(x_{cg})_k = (\cos(\zeta_k)\cos(\beta_k)\cos(\psi_k) - \sin(\zeta_k)\sin(\psi_k))(R_{cg})_k + \cos(\psi_k)e \tag{9.7}$$

$$(y_{cg})_k = (\cos(\zeta_k)\cos(\beta_k)\sin(\psi_k) + \sin(\zeta_k)\cos(\psi_k))(R_{cg})_k + \sin(\psi_k)e \tag{9.8}$$

These time histories contain both regressing and progressing mode contributions. The damping levels of these modes for various gain and phase settings were determined using the moving block analysis program described in a previous section.

Initially, a simulation was run with the feedback gain and phase set to zero in order to get a baseline response. Figure 9.1 shows the results of the baseline response of the rotor center of gravity offset position. Figure 9.2 displays the results of performing a moving block analysis on the x_{cg} signal of Figure 9.1. It can be seen in both figures that the center gravity offset signal contains both the regressing and progressing lead-lag modes. The progressing mode damps out relatively quickly and is of little influence after approximately 0.4 seconds of simulation (it can be seen that the high frequency component of the signals in Figure 9.1 "washes out" by 0.4 seconds).

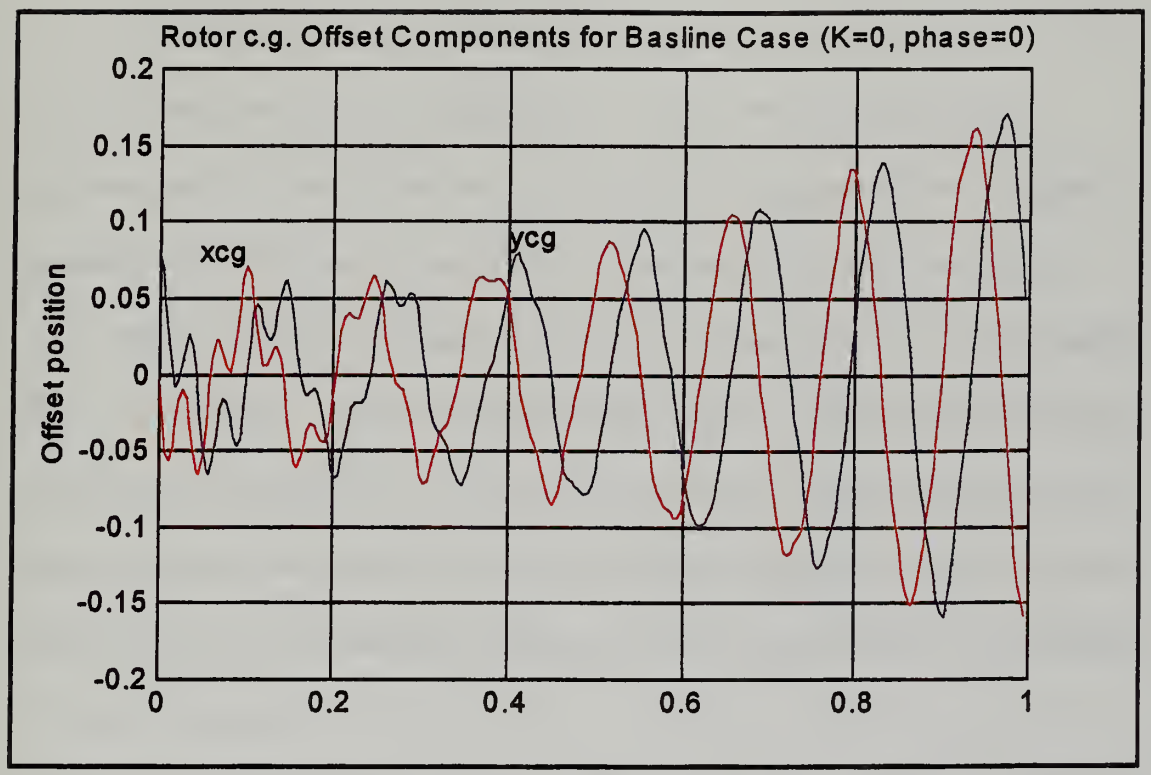


Figure 9.1 Rotor c.g. Offset For Baseline Case (K=0, ϕ =0)

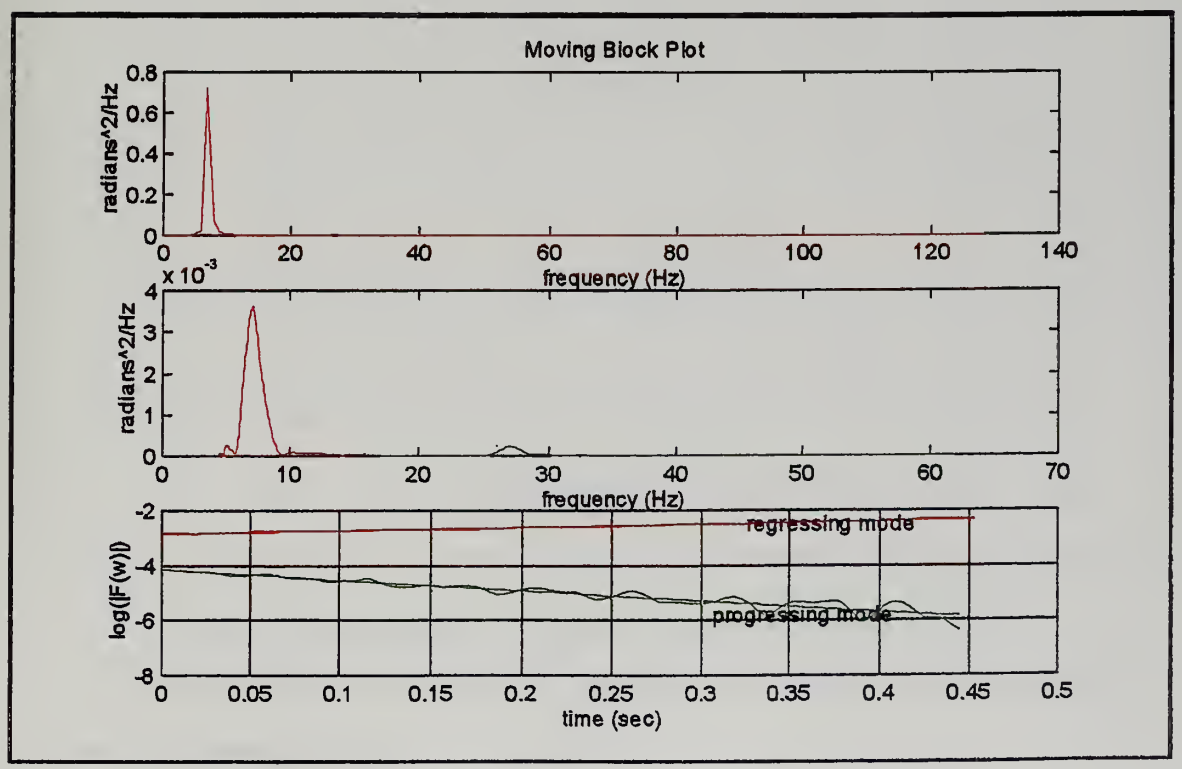


Figure 9.2 Moving Block Results for Baseline Simulation (K=0, ϕ=0)

The moving block analysis determined a damping ratio of -0.0247 (ω=7.1511 Hz) for the regressing mode and a damping ratio of 0.0230 (ω=27.1916 Hz) for the progressing mode. The next step was to conduct the same analysis as was completed on the baseline case for a range of active control phase angles at fixed values of gain.

Figure 9.3 displays the results of running a controller phase sweep at gain settings of $K = 0.4$ and $K=0.8$. The case chosen is slightly unstable, with a no control damping ratio shown by the green line. The results demonstrate that stability can be improved by active swashplate control inputs and that the simulation techniques used in this study can provide a useful tool for predicting which gain and phase combinations would be required for a simple controller.

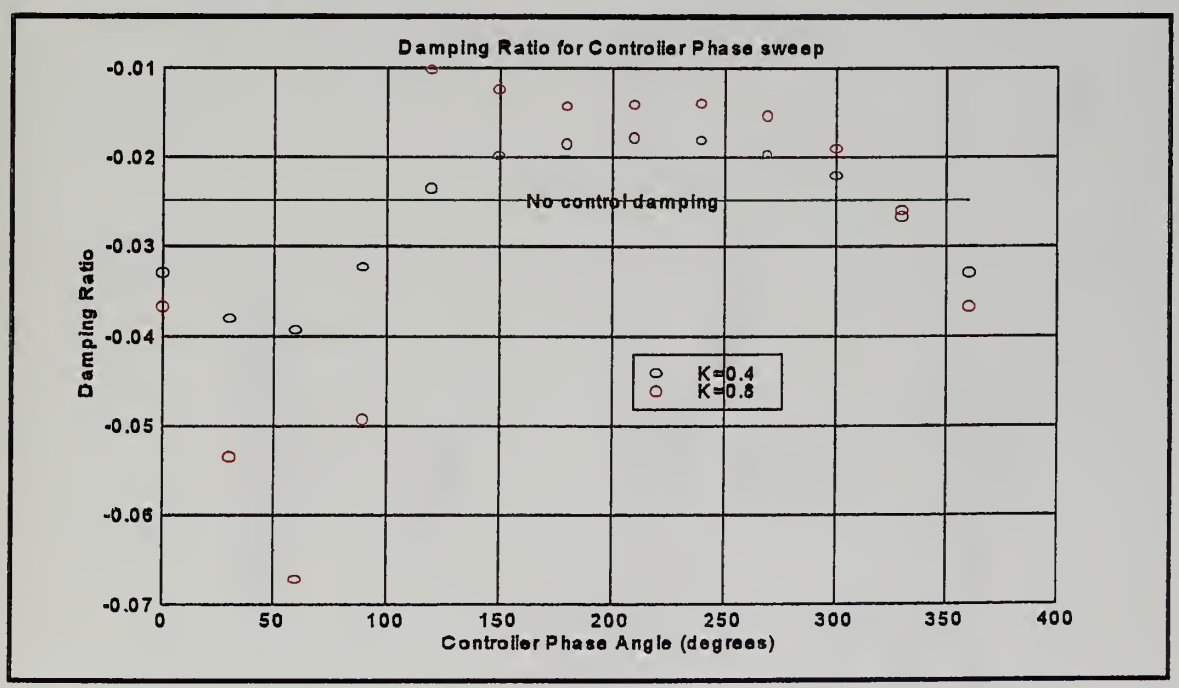
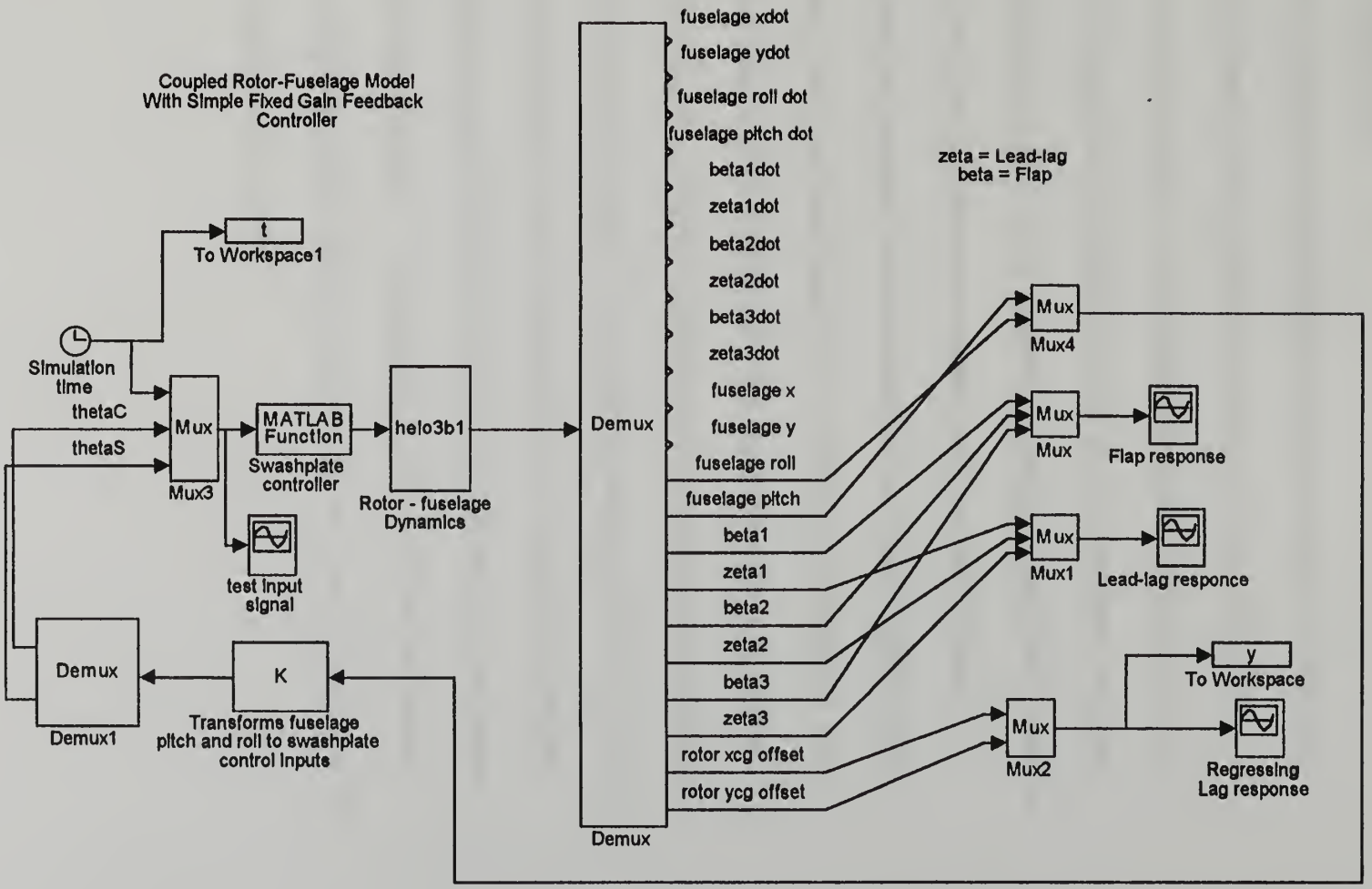


Figure 9.3 Damping Ratio For Controller Phase Sweep (K=0.4 and K=0.8)

Figure 9.4 shows the SIMULINK® model utilized for complex model simulations and the active control analysis.

Figure 9.4 SIMULINK® model implementing fixed feedback gain active rotor control



X. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

A. CONCLUDING REMARKS

A method for formulating and automatically coding the equations of motion of a coupled rotor-fuselage system by use of symbolic processing software and dynamic simulation software has been developed. The resulting mathematical models were used to perform simulations of coupled rotor-fuselage systems in ground resonance. Analysis of the dynamic and stability characteristics were quantified using the moving block technique. A simple rotor model was used to demonstrate essential characteristics of ground resonance and the effects that parameter variations such as rotor speed, flexbeam elastic behavior, damper failure, and rotor blade damage have on those characteristics. A more complex model, adding fuselage pitch and roll and rotor blade flap degrees of freedom, was used to demonstrate how the modeling technique could be used to explore the effect of active rotor control on ground resonance. The modeling technique proved to be a very powerful tool in that it eliminated the time consuming process of manually deriving and coding the very complex equations of motion of a multi-degree of freedom rotor system into a dynamic simulation environment. By integrating SIMULINK® into the process, with its versatility in analyzing dynamic systems, the technique has direct application to the design of modern damperless rotor systems.

B. RECOMMENDATIONS FOR FUTURE RESEARCH

1. Addition of rotor blade torsional degrees of freedom to overall model.
2. Addition of geometric characteristics such as pre-cone, pre-sweep, offset hinge inclination, elastic coupling (pitch-lag, lag-flap, etc.).
3. Addition of unsteady aerodynamics such as a finite state wake model or dynamic inflow model.
4. Incorporation of a trim routine so the model can be used for hover and forward flight aeromechanical stability analysis.
5. A comprehensive study of active control methods using the developed modeling technique, including optimization techniques (such as LQR and LQG) and individual blade control.
6. Validation of simulation results with experiment.

APPENDIX A

MAPLE® WORKSHEET USED TO GENERATE AND CODE THE EQUATIONS OF MOTION FOR A COUPLED ROTOR FUSELAGE SYSTEM

APPENDIX A

(MAPLE WORKSHEET)

EQUATIONS OF MOTION FOR A HELICOPTER IN GROUND RESONANCE CONSIDERING 4 FUSELAGE DEGREES OF FREEDOM AND ROTOR BLADE FLAP AND LEAD-LAG DEGREES OF FREEDOM

```
> restart:
> with(linalg):
Warning, new definition for norm
Warning, new definition for trace
> diff1:=(arg)->map(diff,arg,t):
```

Define coordinate transformations:

```
> psi:=Omega*t+Phi[k];
```

$$\psi := \Omega\, t + \Phi_k$$

```
> T3:=alpha->matrix(3,3,[cos(alpha),sin(alpha),0,-sin(alpha),cos(alpha),0,0,0,1]);
```

$$T3 := \alpha \rightarrow \mathrm{matrix}(3,\,3,\,[\cos(\alpha),\,\sin(\alpha),\,0,\,-\sin(\alpha),\,\cos(\alpha),\,0,\,0,\,0,\,1])$$

```
> T2:=alpha->matrix(3,3,[cos(alpha),0,sin(alpha),0,1,0,-sin(alpha),0,cos(alpha)]);
```

$$T2 := \alpha \rightarrow \mathrm{matrix}(3,\,3,\,[\cos(\alpha),\,0,\,\sin(\alpha),\,0,\,1,\,0,\,-\sin(\alpha),\,0,\,\cos(\alpha)])$$

```
> T1:=alpha->matrix(3,3,[1,0,0,0,cos(alpha),sin(alpha),0,-sin(alpha),cos(alpha)]);
```

$$T1 := \alpha \rightarrow \mathrm{matrix}(3,\,3,\,[1,\,0,\,0,\,0,\,\cos(\alpha),\,\sin(\alpha),\,0,\,-\sin(\alpha),\,\cos(\alpha)])$$

```
> M1:=transpose(multiply(T1(r[1](t)),T2(r[2](t))));
```

$$M1 := \begin{bmatrix} \cos(r_2(t)) & -\sin(r_1(t))\sin(r_2(t)) & -\cos(r_1(t))\sin(r_2(t)) \\ 0 & \cos(r_1(t)) & -\sin(r_1(t)) \\ \sin(r_2(t)) & \sin(r_1(t))\cos(r_2(t)) & \cos(r_1(t))\cos(r_2(t)) \end{bmatrix}$$

```
> M2:=transpose(T3(psi));
```

$$M2 := \begin{bmatrix} \cos(\%1) & -\sin(\%1) & 0 \\ \sin(\%1) & \cos(\%1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\%1 := \Omega\, t + \Phi_k$$

```
> M3:=transpose(multiply(T3(zeta[k](t)),T2(beta[k](t))));
```

$$M3 := \begin{bmatrix} \cos(\zeta_k(t))\cos(\beta_k(t)) & -\sin(\zeta_k(t))\cos(\beta_k(t)) & -\sin(\beta_k(t)) \\ \sin(\zeta_k(t)) & \cos(\zeta_k(t)) & 0 \\ \cos(\zeta_k(t))\sin(\beta_k(t)) & -\sin(\zeta_k(t))\sin(\beta_k(t)) & \cos(\beta_k(t)) \end{bmatrix}$$

```
> M4:=multiply(T3(zeta[k](t)),T2(beta[k](t)),T3(psi),T1(r[1](t)),T2(
  r[2](t))):
```

Energy expressions for kth rotor blade

Kinetic energy of kth blade (TBk)

```
> rhoFI_I:=vector([u[1](t),u[2](t),0]):
> rhoHF:=vector([0,0,h]):
> rhoHF_I:=multiply(M1,rhoHF):
> rhoBuH:=vector([e1,0,0]):
> rhoBuH_I:=multiply(M1,M2,rhoBuH):
> rhoPBd:=vector([R,0,0]):
> rhoPBd_I:=multiply(M1,M2,M3,rhoPBd):
> rho:=matadd(rhoFI_I,matadd(rhoHF_I,matadd(rhoBuH_I,rhoPBd_I))):
> V:=diff1(rho):
> Vsqr:=V[1]^2+V[2]^2+V[3]^2:
> TBk:=1/2*mb[k]*Vsqr;
```

$$
\begin{aligned}
TBk := \frac{1}{2} mb_k \Bigg(& \Bigg(\Bigg(\left(\frac{\partial}{\partial t} u_1(t)\right) + \sin(r_1(t)) \left(\frac{\partial}{\partial t} r_1(t)\right) \sin(r_2(t)) \, h \\
& - \cos(r_1(t)) \cos(r_2(t)) \left(\frac{\partial}{\partial t} r_2(t)\right) h + \left(-\sin(r_2(t)) \left(\frac{\partial}{\partial t} r_2(t)\right) \cos(\%3) - \cos(r_2(t)) \sin(\%3) \, \Omega \right. \\
& - \cos(r_1(t)) \left(\frac{\partial}{\partial t} r_1(t)\right) \sin(r_2(t)) \sin(\%3) - \sin(r_1(t)) \cos(r_2(t)) \left(\frac{\partial}{\partial t} r_2(t)\right) \sin(\%3) \\
& \left. - \sin(r_1(t)) \sin(r_2(t)) \cos(\%3) \, \Omega \right) e1 + \Bigg(\Bigg(-\sin(r_2(t)) \left(\frac{\partial}{\partial t} r_2(t)\right) \cos(\%3) \\
& - \cos(r_2(t)) \sin(\%3) \, \Omega - \cos(r_1(t)) \left(\frac{\partial}{\partial t} r_1(t)\right) \sin(r_2(t)) \sin(\%3) \\
& \left. - \sin(r_1(t)) \cos(r_2(t)) \left(\frac{\partial}{\partial t} r_2(t)\right) \sin(\%3) - \sin(r_1(t)) \sin(r_2(t)) \cos(\%3) \, \Omega \right) \cos(\zeta_k(t)) \\
& \cos(\beta_k(t)) - (\cos(r_2(t)) \cos(\%3) - \sin(r_1(t)) \sin(r_2(t)) \sin(\%3)) \sin(\zeta_k(t)) \, \%2 \cos(\beta_k(t)) \\
& - (\cos(r_2(t)) \cos(\%3) - \sin(r_1(t)) \sin(r_2(t)) \sin(\%3)) \cos(\zeta_k(t)) \sin(\beta_k(t)) \, \%1 + \Bigg(\\
& \sin(r_2(t)) \left(\frac{\partial}{\partial t} r_2(t)\right) \sin(\%3) - \cos(r_2(t)) \cos(\%3) \, \Omega \\
& - \cos(r_1(t)) \left(\frac{\partial}{\partial t} r_1(t)\right) \sin(r_2(t)) \cos(\%3) - \sin(r_1(t)) \cos(r_2(t)) \left(\frac{\partial}{\partial t} r_2(t)\right) \cos(\%3) \\
& \left. + \sin(r_1(t)) \sin(r_2(t)) \sin(\%3) \, \Omega \right) \sin(\zeta_k(t))
\end{aligned}
$$

$$+(-\cos(r_2(t))\sin(\%3)-\sin(r_1(t))\sin(r_2(t))\cos(\%3))\cos(\zeta_k(t))\,\%2$$
$$+\sin(r_1(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\sin(r_2(t))\cos(\zeta_k(t))\sin(\beta_k(t))$$
$$-\cos(r_1(t))\cos(r_2(t))\left(\frac{\partial}{\partial t}r_2(t)\right)\cos(\zeta_k(t))\sin(\beta_k(t))$$
$$+\cos(r_1(t))\sin(r_2(t))\sin(\zeta_k(t))\,\%2\sin(\beta_k(t))$$
$$-\cos(r_1(t))\sin(r_2(t))\cos(\zeta_k(t))\cos(\beta_k(t))\,\%1\Big)R\Big)^2+\left(\left(\left(\frac{\partial}{\partial t}u_2(t)\right)-\cos(r_1(t))\left(\frac{\partial}{\partial t}r_1(t)\right)h\right.\right.$$
$$-\sin(r_1(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\sin(\%3)\,el+\cos(r_1(t))\cos(\%3)\,\Omega\,el+\Bigg($$
$$-\sin(r_1(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\sin(\%3)\cos(\zeta_k(t))\cos(\beta_k(t))$$
$$+\cos(r_1(t))\cos(\%3)\,\Omega\cos(\zeta_k(t))\cos(\beta_k(t))-\cos(r_1(t))\sin(\%3)\sin(\zeta_k(t))\,\%2\cos(\beta_k(t))$$
$$-\cos(r_1(t))\sin(\%3)\cos(\zeta_k(t))\sin(\beta_k(t))\,\%1-\sin(r_1(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\cos(\%3)\sin(\zeta_k(t))$$
$$-\cos(r_1(t))\sin(\%3)\,\Omega\sin(\zeta_k(t))+\cos(r_1(t))\cos(\%3)\cos(\zeta_k(t))\,\%2$$
$$+\sin(\zeta_k(t))\,\%2\sin(\beta_k(t))\sin(r_1(t))-\cos(\zeta_k(t))\cos(\beta_k(t))\,\%1\sin(r_1(t))$$
$$\left.-\cos(\zeta_k(t))\sin(\beta_k(t))\cos(r_1(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\right)R\Big)^2+\left(-\sin(r_1(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\cos(r_2(t))\,h\right.$$
$$-\cos(r_1(t))\sin(r_2(t))\left(\frac{\partial}{\partial t}r_2(t)\right)h+\left(\cos(r_2(t))\left(\frac{\partial}{\partial t}r_2(t)\right)\cos(\%3)-\sin(r_2(t))\sin(\%3)\,\Omega\right.$$
$$+\cos(r_1(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\cos(r_2(t))\sin(\%3)-\sin(r_1(t))\sin(r_2(t))\left(\frac{\partial}{\partial t}r_2(t)\right)\sin(\%3)$$
$$\left.+\sin(r_1(t))\cos(r_2(t))\cos(\%3)\,\Omega\right)el+\Bigg(\Bigg(\left(\cos(r_2(t))\left(\frac{\partial}{\partial t}r_2(t)\right)\cos(\%3)\right.$$
$$-\sin(r_2(t))\sin(\%3)\,\Omega+\cos(r_1(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\cos(r_2(t))\sin(\%3)$$
$$\left.-\sin(r_1(t))\sin(r_2(t))\left(\frac{\partial}{\partial t}r_2(t)\right)\sin(\%3)+\sin(r_1(t))\cos(r_2(t))\cos(\%3)\,\Omega\right)\cos(\zeta_k(t))$$
$$\cos(\beta_k(t))-(\sin(r_2(t))\cos(\%3)+\sin(r_1(t))\cos(r_2(t))\sin(\%3))\sin(\zeta_k(t))\,\%2\cos(\beta_k(t))$$
$$-(\sin(r_2(t))\cos(\%3)+\sin(r_1(t))\cos(r_2(t))\sin(\%3))\cos(\zeta_k(t))\sin(\beta_k(t))\,\%1+\Bigg($$
$$-\cos(r_2(t))\left(\frac{\partial}{\partial t}r_2(t)\right)\sin(\%3)-\sin(r_2(t))\cos(\%3)\,\Omega$$

$$+\cos(r_1(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\cos(r_2(t))\cos(\%3)-\sin(r_1(t))\sin(r_2(t))\left(\frac{\partial}{\partial t}r_2(t)\right)\cos(\%3)$$

$$-\sin(r_1(t))\cos(r_2(t))\sin(\%3)\,\Omega\Big)\sin(\zeta_k(t))$$

$$+(-\sin(r_2(t))\sin(\%3)+\sin(r_1(t))\cos(r_2(t))\cos(\%3))\cos(\zeta_k(t))\,\%2$$

$$-\sin(r_1(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\cos(r_2(t))\cos(\zeta_k(t))\sin(\beta_k(t))$$

$$-\cos(r_1(t))\sin(r_2(t))\left(\frac{\partial}{\partial t}r_2(t)\right)\cos(\zeta_k(t))\sin(\beta_k(t))$$

$$-\cos(r_1(t))\cos(r_2(t))\sin(\zeta_k(t))\,\%2\sin(\beta_k(t))$$

$$+\cos(r_1(t))\cos(r_2(t))\cos(\zeta_k(t))\cos(\beta_k(t))\,\%1\Big)R\Big)^2\Big)$$

$$\%1 := \frac{\partial}{\partial t}\beta_k(t)$$

$$\%2 := \frac{\partial}{\partial t}\zeta_k(t)$$

$$\%3 := \Omega t + \Phi_k$$

Potential energy of kth blade (UBk)

```
> UBk1:=1/2*(beta[k](t)^2*kf1[k]+zeta[k](t)^2*kl1[k]):
> UBk3:=1/4*(beta[k](t)^4*kf3[k]+zeta[k](t)^4*kl3[k]):
> UBk:=UBk1+UBk3;
```

$$UBk := \frac{1}{2}\beta_k(t)^2\,kf1_k + \frac{1}{2}\zeta_k(t)^2\,kl1_k + \frac{1}{4}\beta_k(t)^4\,kf3_k + \frac{1}{4}\zeta_k(t)^4\,kl3_k$$

Dissapative energy of kth blade (DBk)

```
> DBk:=1/2*diff(beta[k](t),t)^2*cf[k]+1/2*diff(zeta[k](t),t)^2*cl[k]
  ;
```

$$DBk := \frac{1}{2}\left(\frac{\partial}{\partial t}\beta_k(t)\right)^2 cf_k + \frac{1}{2}\left(\frac{\partial}{\partial t}\zeta_k(t)\right)^2 cl_k$$

Energy expressions for fuselage

Kinetic energy of fuselage (TF)

```
> TFt:=1/2*(diff(u[1](t),t)^2*M[1]+diff(u[2](t),t)^2*M[2]):
> TFr:=1/2*(diff(r[1](t),t)^2*I11+diff(r[2](t),t)^2*I22-2*diff(r[1](
```

```
   t),t)*diff(r[2](t),t)*I12):
> TF:=TFt+TFr;
```

$$TF := \frac{1}{2}\left(\frac{\partial}{\partial t}u_1(t)\right)^2 M_1 + \frac{1}{2}\left(\frac{\partial}{\partial t}u_2(t)\right)^2 M_2 + \frac{1}{2}\left(\frac{\partial}{\partial t}r_1(t)\right)^2 I11 + \frac{1}{2}\left(\frac{\partial}{\partial t}r_2(t)\right)^2 I22 - \left(\frac{\partial}{\partial t}r_1(t)\right)\left(\frac{\partial}{\partial t}r_2(t)\right) I12$$

Potential energy of fuselage (UF)

```
> UFt:=1/2*u[1](t)^2*KT1+1/2*u[2](t)^2*KT2:
> UFr:=1/2*r[1](t)^2*KR1+1/2*r[2](t)^2*KR2:
> UF:=UFt+UFr;
```

$$UF := \frac{1}{2}u_1(t)^2 KT1 + \frac{1}{2}u_2(t)^2 KT2 + \frac{1}{2}r_1(t)^2 KR1 + \frac{1}{2}r_2(t)^2 KR2$$

Dissapation energy of fuselage (DF)

```
> DFtv:=1/2*diff(u[1](t),t)^2*CT1+1/2*diff(u[2](t),t)^2*CT2:
> DFrv:=1/2*diff(r[1](t),t)^2*CR1+1/2*diff(r[2](t),t)^2*CR2:
> DFth:=1/2*diff(u[1](t),t)^2*abs(diff(u[1](t),t))*VT1+1/2*diff(u[2]
  (t),t)^2*abs(diff(u[2](t),t))*VT2:
> DFrh:=1/2*diff(r[1](t),t)^2*abs(diff(r[1](t),t))*VR1+1/2*diff(r[2]
  (t),t)^2*abs(diff(r[2](t),t))*VR2:
> DF:=DFtv+DFrv+DFth+DFrh;
```

$$DF := \frac{1}{2}\left(\frac{\partial}{\partial t}u_1(t)\right)^2 CT1 + \frac{1}{2}\left(\frac{\partial}{\partial t}u_2(t)\right)^2 CT2 + \frac{1}{2}\left(\frac{\partial}{\partial t}r_1(t)\right)^2 CR1 + \frac{1}{2}\left(\frac{\partial}{\partial t}r_2(t)\right)^2 CR2 + \frac{1}{2}\left(\frac{\partial}{\partial t}u_1(t)\right)^2 \left|\frac{\partial}{\partial t}u_1(t)\right| VT1 + \frac{1}{2}\left(\frac{\partial}{\partial t}u_2(t)\right)^2 \left|\frac{\partial}{\partial t}u_2(t)\right| VT2 + \frac{1}{2}\left(\frac{\partial}{\partial t}r_1(t)\right)^2 \left|\frac{\partial}{\partial t}r_1(t)\right| VR1 + \frac{1}{2}\left(\frac{\partial}{\partial t}r_2(t)\right)^2 \left|\frac{\partial}{\partial t}r_2(t)\right| VR2$$

Aerodynamics (Generalized Aerodynamic Forces)

```
> Vair:=vector([mu*Omega*R,0,Omega*lambda*R]);
```

$$Vair := [\mu\,\Omega\,R,\ 0,\ \Omega\,\lambda\,R]$$

```
> V_I_t:=matadd(-V,Vair):
> V_Bd_t:=map(simplify,multiply(M4,V_I_t)):
> UR:=V_Bd_t[1]:
> UT:=-V_Bd_t[2]:
> UP:=-V_Bd_t[3]:
> UU:=sqrt(UP^2+UT^2):
```

```
> aoa:=theta[k]-arctan(UP/UT):
> dFbeta:=1/2*rho1*a*c*(aoa*UU*UT-cd0/a*UU*UP):
> dFzeta:=-1/2*rho1*a*c*(aoa*UU*UP+cd0/a*UU*UT):
> Mbeta_k:=0.7*R^2*dFbeta*cos(zeta[k](t));
```

$$Mbeta\_k := .3500000000$$

$$R^2\,\rho 1\,a\,c\left(\left(\theta_k - \arctan\left(\frac{\%2}{\%3}\right)\right)\sqrt{\%2^2+\%3^2}\;\%3 - \frac{cd0\sqrt{\%2^2+\%3^2}\;\%2}{a}\right)\cos(\zeta_k(t))$$

$$\%1 := \Omega\,t + \Phi_k$$

$$\%2 := R\cos(\zeta_k(t))\left(\frac{\partial}{\partial t}\beta_k(t)\right) + \cos(r_2(t))\sin(\beta_k(t))\sin(\%1)\sin(r_1(t))\,\Omega\,\lambda\,R$$

$$+ \sin(\beta_k(t))\cos(\%1)\sin(r_2(t))\,\Omega\,\lambda\,R - \sin(\beta_k(t))\sin(\%1)\cos(r_1(t))\left(\frac{\partial}{\partial t}u_2(t)\right)$$

$$- \cos(\beta_k(t))\sin(r_1(t))\left(\frac{\partial}{\partial t}u_2(t)\right) + \sin(r_2(t))\sin(\beta_k(t))\sin(\%1)\sin(r_1(t))\left(\frac{\partial}{\partial t}u_1(t)\right)$$

$$- \sin(\beta_k(t))\cos(\%1)\cos(r_2(t))\left(\frac{\partial}{\partial t}u_1(t)\right) - \sin(r_2(t))\cos(\beta_k(t))\cos(r_1(t))\left(\frac{\partial}{\partial t}u_1(t)\right)$$

$$- \cos(r_2(t))\cos(\beta_k(t))\cos(r_1(t))\,\Omega\,\lambda\,R + \cos(\beta_k(t))\cos(r_1(t))\,el\left(\frac{\partial}{\partial t}r_2(t)\right)\cos(\%1)$$

$$+ \cos(\beta_k(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\sin(\%1)\,el + \cos(\beta_k(t))\,R\left(\frac{\partial}{\partial t}r_1(t)\right)\cos(\%1)\sin(\zeta_k(t))$$

$$+ \sin(\beta_k(t))\sin(\%1)\left(\frac{\partial}{\partial t}r_1(t)\right)h + \sin(\beta_k(t))\cos(\%1)\cos(r_1(t))\left(\frac{\partial}{\partial t}r_2(t)\right)h$$

$$+ \sin(\beta_k(t))\sin(r_1(t))\,R\sin(\zeta_k(t))\left(\frac{\partial}{\partial t}r_2(t)\right) + \cos(\%1)\,R\cos(r_1(t))\left(\frac{\partial}{\partial t}r_2(t)\right)\cos(\zeta_k(t))$$

$$+ \sin(\%1)\,R\left(\frac{\partial}{\partial t}r_1(t)\right)\cos(\zeta_k(t)) + \sin(\beta_k(t))\,R\sin(\zeta_k(t))\,\Omega$$

$$+ \sin(\beta_k(t))\cos(\%1)\cos(r_2(t))\,\mu\,\Omega\,R - \cos(\beta_k(t))\cos(r_1(t))\,R\sin(\zeta_k(t))\left(\frac{\partial}{\partial t}r_2(t)\right)\sin(\%1)$$

$$- \sin(r_2(t))\sin(\beta_k(t))\sin(\%1)\sin(r_1(t))\,\mu\,\Omega\,R + \sin(r_2(t))\cos(\beta_k(t))\cos(r_1(t))\,\mu\,\Omega\,R$$

$$\%3 := \sin(\zeta_k(t))\cos(\beta_k(t))\sin(\%1)\left(\frac{\partial}{\partial t}r_1(t)\right)h + \cos(\zeta_k(t))\,el\,\Omega + \cos(\beta_k(t))\,R\,\Omega$$

$$+ R\left(\frac{\partial}{\partial t}\zeta_k(t)\right) + \sin(\zeta_k(t))\cos(\beta_k(t))\cos(\%1)\cos(r_1(t))\left(\frac{\partial}{\partial t}r_2(t)\right)h$$

$$- \sin(\beta_k(t))\,R\left(\frac{\partial}{\partial t}r_1(t)\right)\cos(\%1) - \sin(\zeta_k(t))\sin(\beta_k(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\sin(\%1)\,el$$

$$+\cos(\zeta_k(t))\,el\,\sin(r_1(t))\left(\frac{\partial}{\partial t}r_2(t)\right)+\cos(\zeta_k(t))\,\sin(\%1)\,\cos(r_1(t))\left(\frac{\partial}{\partial t}r_2(t)\right)h$$

$$-\cos(\zeta_k(t))\,\cos(\%1)\left(\frac{\partial}{\partial t}r_1(t)\right)h-\sin(\zeta_k(t))\,\sin(\beta_k(t))\,\cos(r_1(t))\,el\left(\frac{\partial}{\partial t}r_2(t)\right)\cos(\%1)$$

$$+\sin(\beta_k(t))\,\cos(r_1(t))\,R\left(\frac{\partial}{\partial t}r_2(t)\right)\sin(\%1)+\sin(r_1(t))\,\cos(\beta_k(t))\,R\left(\frac{\partial}{\partial t}r_2(t)\right)$$

$$-\cos(r_2(t))\,\cos(\zeta_k(t))\,\sin(\%1)\left(\frac{\partial}{\partial t}u_1(t)\right)+\cos(r_2(t))\,\sin(\zeta_k(t))\,\cos(\beta_k(t))\,\cos(\%1)\,\mu\,\Omega\,R$$

$$+\cos(r_2(t))\,\cos(\zeta_k(t))\,\sin(\%1)\,\mu\,\Omega\,R$$

$$-\sin(r_2(t))\,\sin(r_1(t))\,\sin(\zeta_k(t))\,\cos(\beta_k(t))\,\sin(\%1)\,\mu\,\Omega\,R$$

$$+\sin(r_2(t))\,\sin(r_1(t))\,\cos(\zeta_k(t))\,\cos(\%1)\,\mu\,\Omega\,R$$

$$-\sin(r_2(t))\,\sin(\zeta_k(t))\,\sin(\beta_k(t))\,\cos(r_1(t))\,\mu\,\Omega\,R$$

$$-\cos(r_2(t))\,\sin(\zeta_k(t))\,\cos(\beta_k(t))\,\cos(\%1)\left(\frac{\partial}{\partial t}u_1(t)\right)$$

$$+\sin(r_2(t))\,\sin(r_1(t))\,\sin(\zeta_k(t))\,\cos(\beta_k(t))\,\sin(\%1)\left(\frac{\partial}{\partial t}u_1(t)\right)$$

$$-\sin(r_2(t))\,\sin(r_1(t))\,\cos(\zeta_k(t))\,\cos(\%1)\left(\frac{\partial}{\partial t}u_1(t)\right)$$

$$+\sin(r_2(t))\,\sin(\zeta_k(t))\,\sin(\beta_k(t))\,\cos(r_1(t))\left(\frac{\partial}{\partial t}u_1(t)\right)$$

$$-\cos(r_1(t))\,\sin(\zeta_k(t))\,\cos(\beta_k(t))\,\sin(\%1)\left(\frac{\partial}{\partial t}u_2(t)\right)$$

$$+\cos(r_1(t))\,\cos(\zeta_k(t))\,\cos(\%1)\left(\frac{\partial}{\partial t}u_2(t)\right)+\sin(\zeta_k(t))\,\sin(\beta_k(t))\,\sin(r_1(t))\left(\frac{\partial}{\partial t}u_2(t)\right)$$

$$+\sin(r_2(t))\,\sin(\zeta_k(t))\,\cos(\beta_k(t))\,\cos(\%1)\,\Omega\,\lambda\,R+\sin(r_2(t))\,\cos(\zeta_k(t))\,\sin(\%1)\,\Omega\,\lambda\,R$$

$$+\cos(r_2(t))\,\sin(r_1(t))\,\sin(\zeta_k(t))\,\cos(\beta_k(t))\,\sin(\%1)\,\Omega\,\lambda\,R$$

$$-\cos(r_2(t))\,\sin(r_1(t))\,\cos(\zeta_k(t))\,\cos(\%1)\,\Omega\,\lambda\,R$$

$$+\cos(r_2(t))\,\sin(\zeta_k(t))\,\sin(\beta_k(t))\,\cos(r_1(t))\,\Omega\,\lambda\,R$$

```
> Mzeta_k:=0.7*R^2*dFzeta;
```

$$Mzeta\_k :=$$

$$-.3500000000\,R^2\,\rho 1\,a\,c\left(\left(\theta_k-\arctan\left(\frac{\%3}{\%2}\right)\right)\sqrt{\%3^2+\%2^2}\;\%3+\frac{cd0\sqrt{\%3^2+\%2^2}\;\%2}{a}\right)$$

$$\%1 := \Omega\,t+\Phi_k$$

$$\%2 := \sin(\zeta_k(t))\,\cos(\beta_k(t))\,\sin(\%1)\left(\frac{\partial}{\partial t}r_1(t)\right)h+\cos(\zeta_k(t))\,el\,\Omega+\cos(\beta_k(t))\,R\,\Omega$$

$$+ R\left(\frac{\partial}{\partial t}\zeta_k(t)\right) + \sin(\zeta_k(t))\cos(\beta_k(t))\cos(\%1)\cos(r_1(t))\left(\frac{\partial}{\partial t}r_2(t)\right)h$$

$$- \sin(\beta_k(t))R\left(\frac{\partial}{\partial t}r_1(t)\right)\cos(\%1) - \sin(\zeta_k(t))\sin(\beta_k(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\sin(\%1)\,el$$

$$+ \cos(\zeta_k(t))\,el\,\sin(r_1(t))\left(\frac{\partial}{\partial t}r_2(t)\right) + \cos(\zeta_k(t))\sin(\%1)\cos(r_1(t))\left(\frac{\partial}{\partial t}r_2(t)\right)h$$

$$- \cos(\zeta_k(t))\cos(\%1)\left(\frac{\partial}{\partial t}r_1(t)\right)h - \sin(\zeta_k(t))\sin(\beta_k(t))\cos(r_1(t))\,el\left(\frac{\partial}{\partial t}r_2(t)\right)\cos(\%1)$$

$$+ \sin(\beta_k(t))\cos(r_1(t))R\left(\frac{\partial}{\partial t}r_2(t)\right)\sin(\%1) + \sin(r_1(t))\cos(\beta_k(t))R\left(\frac{\partial}{\partial t}r_2(t)\right)$$

$$- \cos(r_2(t))\cos(\zeta_k(t))\sin(\%1)\left(\frac{\partial}{\partial t}u_1(t)\right) + \cos(r_2(t))\sin(\zeta_k(t))\cos(\beta_k(t))\cos(\%1)\,\mu\,\Omega\,R$$

$$+ \cos(r_2(t))\cos(\zeta_k(t))\sin(\%1)\,\mu\,\Omega\,R$$

$$- \sin(r_2(t))\sin(r_1(t))\sin(\zeta_k(t))\cos(\beta_k(t))\sin(\%1)\,\mu\,\Omega\,R$$

$$+ \sin(r_2(t))\sin(r_1(t))\cos(\zeta_k(t))\cos(\%1)\,\mu\,\Omega\,R$$

$$- \sin(r_2(t))\sin(\zeta_k(t))\sin(\beta_k(t))\cos(r_1(t))\,\mu\,\Omega\,R$$

$$- \cos(r_2(t))\sin(\zeta_k(t))\cos(\beta_k(t))\cos(\%1)\left(\frac{\partial}{\partial t}u_1(t)\right)$$

$$+ \sin(r_2(t))\sin(r_1(t))\sin(\zeta_k(t))\cos(\beta_k(t))\sin(\%1)\left(\frac{\partial}{\partial t}u_1(t)\right)$$

$$- \sin(r_2(t))\sin(r_1(t))\cos(\zeta_k(t))\cos(\%1)\left(\frac{\partial}{\partial t}u_1(t)\right)$$

$$+ \sin(r_2(t))\sin(\zeta_k(t))\sin(\beta_k(t))\cos(r_1(t))\left(\frac{\partial}{\partial t}u_1(t)\right)$$

$$- \cos(r_1(t))\sin(\zeta_k(t))\cos(\beta_k(t))\sin(\%1)\left(\frac{\partial}{\partial t}u_2(t)\right)$$

$$+ \cos(r_1(t))\cos(\zeta_k(t))\cos(\%1)\left(\frac{\partial}{\partial t}u_2(t)\right) + \sin(\zeta_k(t))\sin(\beta_k(t))\sin(r_1(t))\left(\frac{\partial}{\partial t}u_2(t)\right)$$

$$+ \sin(r_2(t))\sin(\zeta_k(t))\cos(\beta_k(t))\cos(\%1)\,\Omega\,\lambda\,R + \sin(r_2(t))\cos(\zeta_k(t))\sin(\%1)\,\Omega\,\lambda\,R$$

$$+ \cos(r_2(t))\sin(r_1(t))\sin(\zeta_k(t))\cos(\beta_k(t))\sin(\%1)\,\Omega\,\lambda\,R$$

$$- \cos(r_2(t))\sin(r_1(t))\cos(\zeta_k(t))\cos(\%1)\,\Omega\,\lambda\,R$$

$$+ \cos(r_2(t))\sin(\zeta_k(t))\sin(\beta_k(t))\cos(r_1(t))\,\Omega\,\lambda\,R$$

$$\%3 := R\cos(\zeta_k(t))\left(\frac{\partial}{\partial t}\beta_k(t)\right) + \cos(r_2(t))\sin(\beta_k(t))\sin(\%1)\sin(r_1(t))\,\Omega\,\lambda\,R$$

$$+\sin(\beta_k(t))\cos(\%1)\sin(r_2(t))\,\Omega\,\lambda\,R-\sin(\beta_k(t))\sin(\%1)\cos(r_1(t))\left(\frac{\partial}{\partial t}u_2(t)\right)$$

$$-\cos(\beta_k(t))\sin(r_1(t))\left(\frac{\partial}{\partial t}u_2(t)\right)+\sin(r_2(t))\sin(\beta_k(t))\sin(\%1)\sin(r_1(t))\left(\frac{\partial}{\partial t}u_1(t)\right)$$

$$-\sin(\beta_k(t))\cos(\%1)\cos(r_2(t))\left(\frac{\partial}{\partial t}u_1(t)\right)-\sin(r_2(t))\cos(\beta_k(t))\cos(r_1(t))\left(\frac{\partial}{\partial t}u_1(t)\right)$$

$$-\cos(r_2(t))\cos(\beta_k(t))\cos(r_1(t))\,\Omega\,\lambda\,R+\cos(\beta_k(t))\cos(r_1(t))\,el\left(\frac{\partial}{\partial t}r_2(t)\right)\cos(\%1)$$

$$+\cos(\beta_k(t))\left(\frac{\partial}{\partial t}r_1(t)\right)\sin(\%1)\,el+\cos(\beta_k(t))\,R\left(\frac{\partial}{\partial t}r_1(t)\right)\cos(\%1)\sin(\zeta_k(t))$$

$$+\sin(\beta_k(t))\sin(\%1)\left(\frac{\partial}{\partial t}r_1(t)\right)h+\sin(\beta_k(t))\cos(\%1)\cos(r_1(t))\left(\frac{\partial}{\partial t}r_2(t)\right)h$$

$$+\sin(\beta_k(t))\sin(r_1(t))\,R\sin(\zeta_k(t))\left(\frac{\partial}{\partial t}r_2(t)\right)+\cos(\%1)\,R\cos(r_1(t))\left(\frac{\partial}{\partial t}r_2(t)\right)\cos(\zeta_k(t))$$

$$+\sin(\%1)\,R\left(\frac{\partial}{\partial t}r_1(t)\right)\cos(\zeta_k(t))+\sin(\beta_k(t))\,R\sin(\zeta_k(t))\,\Omega$$

$$+\sin(\beta_k(t))\cos(\%1)\cos(r_2(t))\,\mu\,\Omega\,R-\cos(\beta_k(t))\cos(r_1(t))\,R\sin(\zeta_k(t))\left(\frac{\partial}{\partial t}r_2(t)\right)\sin(\%1)$$

$$-\sin(r_2(t))\sin(\beta_k(t))\sin(\%1)\sin(r_1(t))\,\mu\,\Omega\,R+\sin(r_2(t))\cos(\beta_k(t))\cos(r_1(t))\,\mu\,\Omega\,R$$

Derivation of equations of motion by Lagrangian method

This section defines vectors of displacement degrees of freedom, their rates and accelerations.

```
> DOFF:=[u[1](t),u[2](t),r[1](t),r[2](t)]:
```

```
> N:=3;
```
Choose number of rotor blades

$$N := 3$$

```
> DOFB:=[]:ThetaB:=[]:
> for i from 1 to N do
>    DOFB:=[op(DOFB),beta[i](t),zeta[i](t)]:
> od:
```

```
> DOF:=[op(DOFF),op(DOFB)];
```

$$DOF := [u_1(t), u_2(t), r_1(t), r_2(t), \beta_1(t), \zeta_1(t), \beta_2(t), \zeta_2(t), \beta_3(t), \zeta_3(t)]$$

```
> dDOF:=diff1(DOF);
```

$$dDOF := \left[\frac{\partial}{\partial t}u_1(t), \frac{\partial}{\partial t}u_2(t), \frac{\partial}{\partial t}r_1(t), \frac{\partial}{\partial t}r_2(t), \frac{\partial}{\partial t}\beta_1(t), \frac{\partial}{\partial t}\zeta_1(t), \frac{\partial}{\partial t}\beta_2(t), \frac{\partial}{\partial t}\zeta_2(t), \frac{\partial}{\partial t}\beta_3(t), \frac{\partial}{\partial t}\zeta_3(t)\right]$$

```
> ddDOF:=diff1(dDOF);
```

$$ddDOF := \left[\frac{\partial^2}{\partial t^2} u_1(t), \frac{\partial^2}{\partial t^2} u_2(t), \frac{\partial^2}{\partial t^2} r_1(t), \frac{\partial^2}{\partial t^2} r_2(t), \frac{\partial^2}{\partial t^2} \beta_1(t), \frac{\partial^2}{\partial t^2} \zeta_1(t), \frac{\partial^2}{\partial t^2} \beta_2(t), \frac{\partial^2}{\partial t^2} \zeta_2(t), \frac{\partial^2}{\partial t^2} \beta_3(t), \frac{\partial^2}{\partial t^2} \zeta_3(t) \right]$$

This section defines transformations between time dependent and independent notation in terms of substitution sets.

```
> setA:={}:setB:={}:setC:={}:
> setD:={}:setE:={}:setF:={}:
> DOFq:=[]:dDOFq:=[]:ddDOFq:=[]:
> for i from 1 to vectdim(DOF) do
>   DOFq:=[op(DOFq),q[i]]:
>   dDOFq:=[op(dDOFq),dq[i]]:
>   ddDOFq:=[op(ddDOFq),ddq[i]]:
>   setA:=setA union {ddDOF[i]=ddDOFq[i]}:
>   setB:=setB union {dDOF[i]=dDOFq[i]}:
>   setC:=setC union {DOF[i]=DOFq[i]}:
>   setD:=setD union {ddDOFq[i]=ddDOF[i]}:
>   setE:=setE union {dDOFq[i]=dDOF[i]}:
>   setF:=setF union {DOFq[i]=DOF[i]}:
> od:
> set1:=setA union setB union setC:
```

Substitution set to go from dependent to independent

```
> set2:=setD union setE union setF:
```

Substitution set to go from independent to dependent

This section combines all of contributions to the terms of the Lagrange equation.

```
> T:=TF:
> U:=UF:
> D1:=DF:
> GF:=[0,0,0,0]:
> for i from 1 to N do
>    T:=T+subs(k=i,TBk):
>    U:=U+subs(k=i,UBk):
>    D1:=D1+subs(k=i,DBk):
>    GF:=[op(GF),subs(k=i,Mbeta_k),subs(k=i,Mzeta_k)]:
> od:
```

This section carries out the differentiation operation of the Lagrange equation one term at a time

```
> Temp:=subs(set1,T):
> for i from 1 to vectdim(DOF) do
>     temp1:=diff(Temp,dDOFq[i]):
>     temp2:=subs(set2,temp1):
>     temp3:=diff(temp2,t):
>     L1:=subs(set1,temp3):
>     L2:=diff(Temp,DOFq[i]):
```

```
> L3:=diff(subs(set1,U),DOFq[i]):
> L4:=diff(subs(set1,D1),dDOFq[i]):
> GFq:=subs(set1,GF[i]):
> EOM[i]:=L1-L2+L3+L4-GFq:
> od:
```

This section formats the equations of motion into the form A d2x/dt2 = f

```
> A:=matrix(vectdim(DOF),vectdim(DOF));
```

$$A := \text{array}(1 .. 10, 1 .. 10, [\ ])$$

```
> for i from 1 to vectdim(DOF) do
>    for j from 1 to vectdim(DOF) do
>       A[i,j]:=coeff(EOM[i],ddDOFq[j]):
>    od:
> od:
> setZ:={}:
> for i from 1 to vectdim(ddDOFq) do
>    setZ:=setZ union {ddDOFq[i]=0}:
> od:
> f:=array(1..vectdim(DOF));
```

$$f := \text{array}(1 .. 10, [\ ])$$

```
> for i from 1 to vectdim(DOF) do
>    f[i]:=-eval(subs(setZ,EOM[i])):
> od:
```

This section makes a change of notation so equations are compatible with standard MATLAB notation for state variables and inputs

```
> x1:=[]:x1dot:=[]:
> for i from 1 to vectdim(DOF) do x1dot:=[op(x1dot),x[i]] od:
> for i from vectdim(DOF)+1 to 2*vectdim(DOF) do x1:=[op(x1),x[i]]
  od:
> setX:={}:
> for i from 1 to vectdim(DOF) do
>    setX:=setX union {dDOFq[i]=x1dot[i]}:
>    setX:=setX union {DOFq[i]=x1[i]}:
> od:
> setX;
```

$$\{q_8 = x_{18}, dq_8 = x_8, q_6 = x_{16}, dq_7 = x_7, q_7 = x_{17}, dq_9 = x_9, q_9 = x_{19}, dq_{10} = x_{10}, q_{10} = x_{20}, dq_5 = x_5,$$
$$q_5 = x_{15}, dq_6 = x_6, q_2 = x_{12}, dq_3 = x_3, q_3 = x_{13}, dq_4 = x_4, q_4 = x_{14}, dq_1 = x_1, q_1 = x_{11}, dq_2 = x_2\}$$

```
> setX1:={abs(1,x[1])=0,abs(1,x[2])=0,abs(1,x[3])=0,abs(1,x[4])=0};
```

$$setX1 := \{\text{abs}(1, x_3) = 0, \text{abs}(1, x_4) = 0, \text{abs}(1, x_1) = 0, \text{abs}(1, x_2) = 0\}$$

```
> A1:=subs(setX,op(A)):
> f1:=subs(setX,op(f)):
> f2:=subs(setX1,op(f1)):
> B:=augment(A1,f2):
```

```
> readlib(fortran);
```

proc(x) ... end

```
> #fortran(B,optimized);
```

This statement converts equations to computer code

APPENDIX B

MAPLE® GENERATED EQUATIONS OF MOTION FOR SIMPLE THREE BLADED ROTOR-FUSELAGE MODEL

APPENDIX B

Equations of Motion Generated for a Three Bladed Simplified Rotor Model with MAPLE

The following is an excerpt from the MAPLE worksheet which was programmed to carry out the Lagrangian derivation of the equations of motion for a 3 bladed coupled rotor fuselage system.

>

FUSELAGE X-DIRECTION:

```
> EOM1[1]=0;
```

$$
\begin{aligned}
&-mb_1 \cos(\%5)\,\Omega^2\,e1 - mb_2 \cos(\%4)\,\Omega^2\,e1 - mb_3 \cos(\%2)\,\Omega^2\,e1 + M_1\,\%7 \\
&\quad - mb_1\,R \cos(\%5) \cos(\zeta_1(t))\,\%6^2 + mb_2\,\%7 + mb_1\,\%7 + mb_3\,\%7 + c_1 \left(\frac{\partial}{\partial t} u_1(t)\right) \\
&\quad + 2\,mb_1\,R \sin(\%5)\,\Omega \sin(\zeta_1(t))\,\%6 - mb_1\,R \cos(\%5) \sin(\zeta_1(t)) \left(\frac{\partial^2}{\partial t^2} \zeta_1(t)\right) \\
&\quad + mb_1\,R \sin(\%5)\,\Omega^2 \sin(\zeta_1(t)) - 2\,mb_1\,R \cos(\%5)\,\Omega \cos(\zeta_1(t))\,\%6 \\
&\quad + mb_1\,R \sin(\%5) \sin(\zeta_1(t))\,\%6^2 + K_1\,u_1(t) - mb_1\,R \sin(\%5) \cos(\zeta_1(t)) \left(\frac{\partial^2}{\partial t^2} \zeta_1(t)\right) \\
&\quad - mb_1\,R \cos(\%5)\,\Omega^2 \cos(\zeta_1(t)) - mb_2\,R \cos(\%4)\,\Omega^2 \cos(\zeta_2(t)) \\
&\quad + 2\,mb_2\,R \sin(\%4)\,\Omega \sin(\zeta_2(t))\,\%3 - mb_2\,R \cos(\%4) \cos(\zeta_2(t))\,\%3^2 \\
&\quad - mb_2\,R \cos(\%4) \sin(\zeta_2(t)) \left(\frac{\partial^2}{\partial t^2} \zeta_2(t)\right) + mb_2\,R \sin(\%4)\,\Omega^2 \sin(\zeta_2(t)) \\
&\quad - 2\,mb_2\,R \cos(\%4)\,\Omega \cos(\zeta_2(t))\,\%3 + mb_2\,R \sin(\%4) \sin(\zeta_2(t))\,\%3^2 \\
&\quad - mb_2\,R \sin(\%4) \cos(\zeta_2(t)) \left(\frac{\partial^2}{\partial t^2} \zeta_2(t)\right) - mb_3\,R \cos(\%2)\,\Omega^2 \cos(\zeta_3(t)) \\
&\quad + 2\,mb_3\,R \sin(\%2)\,\Omega \sin(\zeta_3(t))\,\%1 - mb_3\,R \cos(\%2) \cos(\zeta_3(t))\,\%1^2 \\
&\quad - mb_3\,R \cos(\%2) \sin(\zeta_3(t)) \left(\frac{\partial^2}{\partial t^2} \zeta_3(t)\right) + mb_3\,R \sin(\%2)\,\Omega^2 \sin(\zeta_3(t)) \\
&\quad - 2\,mb_3\,R \cos(\%2)\,\Omega \cos(\zeta_3(t))\,\%1 + mb_3\,R \sin(\%2) \sin(\zeta_3(t))\,\%1^2 \\
&\quad - mb_3\,R \sin(\%2) \cos(\zeta_3(t)) \left(\frac{\partial^2}{\partial t^2} \zeta_3(t)\right) = 0
\end{aligned}
$$

$$\%1 := \frac{\partial}{\partial t}\zeta_3(t)$$

$$\%2 := \Omega\, t + \Phi_3$$

$$\%3 := \frac{\partial}{\partial t}\zeta_2(t)$$

$$\%4 := \Omega\, t + \Phi_2$$

$$\%5 := \Omega\, t + \Phi_1$$

$$\%6 := \frac{\partial}{\partial t}\zeta_1(t)$$

$$\%7 := \frac{\partial^2}{\partial t^2}u_1(t)$$

FUSELAGE Y-DIRECTION:

```
> EOM1[2]=0;
```

$$-mb_3 \sin(\%6)\,\Omega^2\, e1 - mb_1 \sin(\%4)\,\Omega^2\, e1 - mb_2 \sin(\%2)\,\Omega^2\, e1 + c_2\left(\frac{\partial}{\partial t}u_2(t)\right) + K_2\, u_2(t)$$

$$+ M_2\,\%7 + mb_1\,\%7 + mb_2\,\%7 + mb_3\,\%7 - mb_2\, R \sin(\%2)\sin(\zeta_2(t))\left(\frac{\partial^2}{\partial t^2}\zeta_2(t)\right)$$

$$- mb_2\, R \cos(\%2)\,\Omega^2 \sin(\zeta_2(t)) - 2\, mb_2\, R \sin(\%2)\,\Omega \cos(\zeta_2(t))\,\%1$$

$$- mb_2\, R \cos(\%2)\sin(\zeta_2(t))\,\%1^2 + mb_3\, R \cos(\%6)\cos(\zeta_3(t))\left(\frac{\partial^2}{\partial t^2}\zeta_3(t)\right)$$

$$- mb_3\, R \sin(\%6)\,\Omega^2 \cos(\zeta_3(t)) - 2\, mb_3\, R \cos(\%6)\,\Omega \sin(\zeta_3(t))\,\%5$$

$$- mb_3\, R \sin(\%6)\cos(\zeta_3(t))\,\%5^2 - mb_3\, R \sin(\%6)\sin(\zeta_3(t))\left(\frac{\partial^2}{\partial t^2}\zeta_3(t)\right)$$

$$- mb_3\, R \cos(\%6)\,\Omega^2 \sin(\zeta_3(t)) - 2\, mb_3\, R \sin(\%6)\,\Omega \cos(\zeta_3(t))\,\%5$$

$$- mb_3\, R \cos(\%6)\sin(\zeta_3(t))\,\%5^2 - mb_1\, R \sin(\%4)\,\Omega^2 \cos(\zeta_1(t))$$

$$- 2\, mb_1\, R \cos(\%4)\,\Omega \sin(\zeta_1(t))\,\%3 - mb_1\, R \sin(\%4)\cos(\zeta_1(t))\,\%3^2$$

$$- mb_1\, R \sin(\%4)\sin(\zeta_1(t))\left(\frac{\partial^2}{\partial t^2}\zeta_1(t)\right) - mb_1\, R \cos(\%4)\,\Omega^2 \sin(\zeta_1(t))$$

$$- 2\, mb_1\, R \sin(\%4)\,\Omega \cos(\zeta_1(t))\,\%3 - mb_1\, R \cos(\%4)\sin(\zeta_1(t))\,\%3^2$$

$$+ mb_1\, R \cos(\%4)\cos(\zeta_1(t))\left(\frac{\partial^2}{\partial t^2}\zeta_1(t)\right) - mb_2\, R \sin(\%2)\,\Omega^2 \cos(\zeta_2(t))$$

$$- 2\, mb_2\, R \cos(\%2)\,\Omega \sin(\zeta_2(t))\,\%1 + mb_2\, R \cos(\%2)\cos(\zeta_2(t))\left(\frac{\partial^2}{\partial t^2}\zeta_2(t)\right)$$

$$- mb_2\, R \sin(\%2)\cos(\zeta_2(t))\,\%1^2 = 0$$

$$\%1 := \frac{\partial}{\partial t}\zeta_2(t)$$

$$\%2 := \Omega\, t + \Phi_2$$

$$\%3 := \frac{\partial}{\partial t}\zeta_1(t)$$

$$\%4 := \Omega\, t + \Phi_1$$

$$\%5 := \frac{\partial}{\partial t}\zeta_3(t)$$

$$\%6 := \Omega\, t + \Phi_3$$

$$\%7 := \frac{\partial^2}{\partial t^2}u_2(t)$$

ROTOR BLADE 1 LEAD LAG:

```
> EOM1[3]=0;
```

$$Ke_1\,\zeta_1(t) + Czeta_1\left(\frac{\partial}{\partial t}\zeta_1(t)\right) - mb_1\left(\frac{\partial^2}{\partial t^2}u_1(t)\right)R\cos(\%1)\sin(\zeta_1(t))$$
$$- mb_1\left(\frac{\partial^2}{\partial t^2}u_1(t)\right)R\sin(\%1)\cos(\zeta_1(t)) + mb_1\, R^2\left(\frac{\partial^2}{\partial t^2}\zeta_1(t)\right)$$
$$- mb_1\left(\frac{\partial^2}{\partial t^2}u_2(t)\right)R\sin(\%1)\sin(\zeta_1(t)) + mb_1\left(\frac{\partial^2}{\partial t^2}u_2(t)\right)R\cos(\%1)\cos(\zeta_1(t))$$
$$+ mb_1\,\Omega^2\, e1\, R\sin(\zeta_1(t)) = 0$$

$$\%1 := \Omega\, t + \Phi_1$$

ROTOR BLADE 2 LEAD LAG:

```
> EOM1[4]=0;
```

$$Ke_2\,\zeta_2(t) + mb_2\,\Omega^2\, e1\, R\sin(\zeta_2(t)) + mb_2\, R^2\left(\frac{\partial^2}{\partial t^2}\zeta_2(t)\right) - mb_2\left(\frac{\partial^2}{\partial t^2}u_2(t)\right)R\sin(\%1)\sin(\zeta_2(t))$$
$$+ mb_2\left(\frac{\partial^2}{\partial t^2}u_2(t)\right)R\cos(\%1)\cos(\zeta_2(t)) - mb_2\left(\frac{\partial^2}{\partial t^2}u_1(t)\right)R\cos(\%1)\sin(\zeta_2(t))$$
$$- mb_2\left(\frac{\partial^2}{\partial t^2}u_1(t)\right)R\sin(\%1)\cos(\zeta_2(t)) + Czeta_2\left(\frac{\partial}{\partial t}\zeta_2(t)\right) = 0$$

$$\%1 := \Omega\, t + \Phi_2$$

ROTOR BLADE 3 LEAD LAG:

```
> EOM1[5]=0;
```

$$
\begin{aligned}
&Ke_3\,\zeta_3(t) + Czeta_3\left(\frac{\partial}{\partial t}\zeta_3(t)\right) + mb_3\,R^2\left(\frac{\partial^2}{\partial t^2}\zeta_3(t)\right) - mb_3\left(\frac{\partial^2}{\partial t^2}u_2(t)\right)R\sin(\%1)\sin(\zeta_3(t)) \\
&\quad + mb_3\left(\frac{\partial^2}{\partial t^2}u_2(t)\right)R\cos(\%1)\cos(\zeta_3(t)) - mb_3\left(\frac{\partial^2}{\partial t^2}u_1(t)\right)R\cos(\%1)\sin(\zeta_3(t)) \\
&\quad - mb_3\left(\frac{\partial^2}{\partial t^2}u_1(t)\right)R\sin(\%1)\cos(\zeta_3(t)) + mb_3\,\Omega^2\,e1\,R\sin(\zeta_3(t)) = 0 \\
&\%1 := \Omega\,t + \Phi_3
\end{aligned}
$$

APPENDIX C

OPTIMIZED CODE GENERATED BY MAPLE® FOR THE SIMPLE ROTOR-FUSELAGE THREE BLADED MODEL

APPENDIX C

OPTIMIZED CODE GENERATED BY MAPLE FOR THE SIMPLE THREE BLADED COUPLED ROTOR- FUSELAGE MODEL

MAPLE converts the elements of B, which is an augmented matrix B= [A f], from their symbolic representation into FORTRAN code (or C code if desired).

```
> fortran(B,optimized);
      t2 = mb(1)*R
      t3 = Omega*t
      t4 = t3+Phi(1)
      t5 = cos(t4)
      t6 = sin(x(8))
      t7 = t5*t6
      t9 = sin(t4)
      t10 = cos(x(8))
      t11 = t9*t10
      t13 = -t2*t7-t2*t11
      t14 = mb(2)*R
      t15 = t3+Phi(2)
      t16 = cos(t15)
      t17 = sin(x(9))
      t18 = t16*t17
      t20 = sin(t15)
      t21 = cos(x(9))
      t22 = t20*t21
      t24 = -t14*t18-t14*t22
      t25 = mb(3)*R
      t26 = t3+Phi(3)
      t27 = cos(t26)
      t28 = sin(x(10))
      t29 = t27*t28
      t31 = sin(t26)
      t32 = cos(x(10))
      t33 = t31*t32
      t35 = -t25*t29-t25*t33
      t42 = Omega**2
      t43 = t42*e1
      t45 = t5*t42
      t48 = t2*t9
      t50 = Omega*t6*x(3)
      t52 = t5*t10
      t53 = x(3)**2
      t56 = t9*t42
      t59 = t2*t5
      t61 = Omega*t10*x(3)
      t63 = t9*t6
      t68 = t16*t42
      t71 = -v(1)*x(1)*abs(x(1))-K(1)*x(6)-c(1)*x(1)+mb(1)*t5*t43+t2*t45
     #*t10-2*t48*t50+t2*t52*t53-t2*t56*t6+2*t59*t61-t2*t63*t53+mb(2)*t16
     #*t43+t14*t68*t21
      t72 = t14*t20
      t74 = Omega*t17*x(4)
      t76 = t16*t21
```

```
 t77 = x(4)**2
 t80 = t20*t42
 t83 = t14*t16
 t85 = Omega*t21*x(4)
 t87 = t20*t17
 t92 = t27*t42
 t95 = t25*t31
 t97 = Omega*t28*x(5)
 t99 = t27*t32
 t100 = x(5)**2
 t103 = t31*t42
 t106 = t25*t27
 t108 = Omega*t32*x(5)
 t110 = t31*t28
 t113 = -2*t72*t74+t14*t76*t77-t14*t80*t17+2*t83*t85-t14*t87*t77+mb
#(3)*t27*t43+t25*t92*t32-2*t95*t97+t25*t99*t100-t25*t103*t28+2*t106
#*t108-t25*t110*t100
 t118 = -t2*t63+t2*t52
 t121 = -t14*t87+t14*t76
 t124 = t25*t99-t25*t110
 t145 = -c(2)*x(2)-v(2)*x(2)*abs(x(2))-K(2)*x(7)+t25*t92*t28+2*t95*
#t108+mb(1)*t9*t43+t2*t56*t10+2*t59*t50+t2*t11*t53+t2*t45*t6+2*t48*
#t61+t2*t7*t53
 t167 = mb(2)*t20*t43+t25*t29*t100+t14*t80*t21+2*t83*t74+t14*t22*t7
#7+t14*t68*t17+2*t72*t85+t14*t18*t77+mb(3)*t31*t43+t25*t103*t32+2*t
#106*t97+t25*t33*t100
 t169 = R**2
 t176 = Ks(1)*signum(x(8)-z)
 t180 = Ks(1)*signum(x(8)+z)
 t185 = e1*R
 t189 = x(8)**2
 t194 = u(1)-2*Vzeta(1)*x(3)*abs(x(3))-t176*x(8)/2+t180*z/2+t180*x(
#8)/2+t176*z/2-mb(1)*t42*t185*t6-Ke(1)*x(8)-Kd(1)*t189*x(8)-Ks(1)*x
#(8)-Czeta(1)*x(3)
 t198 = Ks(2)*signum(x(9)-z)
 t203 = Ks(2)*signum(x(9)+z)
 t213 = x(9)**2
 t218 = t198*z/2-t198*x(9)/2+t203*z/2+t203*x(9)/2-2*Vzeta(2)*x(4)*a
#bs(x(4))+u(2)-mb(2)*t42*t185*t17-Ke(2)*x(9)-Kd(2)*t213*x(9)-Ks(2)*
#x(9)-Czeta(2)*x(4)
 t220 = x(10)**2
 t226 = Ks(3)*signum(x(10)-z)
 t231 = Ks(3)*signum(x(10)+z)
 t242 = -Kd(3)*t220*x(10)-Ke(3)*x(10)+u(3)-t226*x(10)/2+t226*z/2+t2
#31*x(10)/2+t231*z/2-2*Vzeta(3)*x(5)*abs(x(5))-mb(3)*t42*t185*t28-C
#zeta(3)*x(5)-Ks(3)*x(10)
 B(1,1) = mb(1)+mb(2)+mb(3)+M(1)
 B(1,2) = 0
 B(1,3) = t13
 B(1,4) = t24
 B(1,5) = t35
 B(1,6) = t71+t113
 B(2,1) = 0
 B(2,2) = M(2)+mb(1)+mb(2)+mb(3)
 B(2,3) = t118
 B(2,4) = t121
 B(2,5) = t124
```

```
B(2,6) = t145+t167
B(3,1) = t13
B(3,2) = t118
B(3,3) = mb(1)*t169
B(3,4) = 0
B(3,5) = 0
B(3,6) = t194
B(4,1) = t24
B(4,2) = t121
B(4,3) = 0
B(4,4) = mb(2)*t169
B(4,5) = 0
B(4,6) = t218
B(5,1) = t35
B(5,2) = t124
B(5,3) = 0
B(5,4) = 0
B(5,5) = mb(3)*t169
B(5,6) = t242
```

APPENDIX D

S-FUNCTION M-FILE REPRESENTING THE DYNAMICS OF THE SIMPLE ROTOR-FUSELAGE THREE BLADED MODEL

```
function [sys, x0] = helo3bA(t,x,u,flag,I1,I2,I3,I4,I5,I6)

% function [sys, x0] = helo3bA(t,x,u,flag,I1,I2,I3,I4,I5,I6)
%
%  S-function arguments:
%  ---------------------
%  t     = time
%  x     = state vector
%  u     = input vector
%  flag  = switch used by numerical integration (simulation)
%          routine to access certain parts of the s-function
%
%  S-function input parameters:
%  ----------------------------
%
%  I1    = [mb(1),mb(2),mb(3),M(1),M(2)]
%
%  I2    = [R,Omega,e1,z]
%
%  I3    = [Phi(1),Phi(2),Phi(3)]
%
%  I4    = [c(1),c(2),v(1),v(2),
%           Czeta(1),Czeta(2),Czeta(3),
%           Vzeta(1),Vzeta(2),Vzeta(3)]
%
%  I5    = [Ke(1),Ke(2),Ke(3),
%           Kd(1),Kd(2),Kd(3),
%           Ks(1),Ks(2),Ks(3),
%           K(1),K(2)]
%
%  I6    = [xrXi,xrYi,xr1i,xr2i,xr3i,
%           xXi,xYi,x1i,x2i,x3i]
%
% S-function to represent dynamics of 3 bladed coupled rotor-
% fuselage model which considers only inplane degrees of
% freedom, i.e., x and y translational fuselage degrees of freedom
% and lead-lag rotor blade degrees of freedom.
```

```
%
% Explanation of variables:
% --------------------
%
% mb     ->   mass of blade
% M      ->   effective mass of fuselage
% R      ->   distance from lead-lag hinge to blade center of mass
% e1     ->   blade hinge offset
% Omega  ->   rotor speed
% z      ->   angle at which blade hits stops
% Phi    ->   blade phase angle w.r.t. azimuth postion
% c      ->   fuselage linear damping
% v      ->   fuselage hydraulic damping
% Czeta  ->   blade linear damping
% Vzeta  ->   blade hydraulic damping
% K      ->   effective stiffness of fuselage (landing gear stiffness)
% Ke     ->   blade elastic spring constant
% Kd     ->   blade duffing spring constant
% Ks     ->   blade stop effective spring constant
% xr__i  ->   initial rate
% x__i   ->   initial displacement
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Define input parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mb(1)=I1(1);mb(2)=I1(2);mb(3)=I1(3);M(1)=I1(4);M(2)=I1(5);
R=I2(1);Omega=I2(2);e1=I2(3);z=I2(4);
Phi(1)=I3(1);Phi(2)=I3(2);Phi(3)=I3(3);
c(1)=I4(1);c(2)=I4(2);v(1)=I4(3);v(2)=I4(4);
Czeta(1)=I4(5);Czeta(2)=I4(6);Czeta(3)=I4(7);
Vzeta(1)=I4(8);Vzeta(2)=I4(9);Vzeta(3)=I4(10);
Ke(1)=I5(1);Ke(2)=I5(2);Ke(3)=I5(3);
Kd(1)=I5(4);Kd(2)=I5(5);Kd(6)=I5(6);
Ks(1)=I5(7);Ks(2)=I5(8);Ks(3)=I5(9);
K(1)=I5(10);K(2)=I5(11);
xrXi=I6(1);xrYi=I6(2);xr1i=I6(3);xr2i=I6(4);xr3i=I6(5);
xXi=I6(6);xYi=I6(7);x1i=I6(8);x2i=I6(9);x3i=I6(10);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% S-function flag conditionals
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if flag == 0

    sys=[10,0,10,3,0,0];
```

```
    x0=[xrXi,xrYi,xr1i,xr2i,xr3i,xXi,xYi,x1i,x2i,x3i];

elseif abs(flag) == 1

% Formulated equations of motion optimized for minimum number of floating
% point operations.

    t2 = mb(1)*R;
    t3 = Omega*t;
    t4 = t3+Phi(1);
    t5 = sin(t4);
    t6 = cos(x(8));
    t7 = t5*t6;
    t9 = cos(t4);
    t10 = sin(x(8));
    t11 = t9*t10;
    t13 = -t2*t7-t2*t11;
    t14 = mb(2)*R;
    t15 = t3+Phi(2);
    t16 = cos(t15);
    t17 = sin(x(9));
    t18 = t16*t17;
    t20 = sin(t15);
    t21 = cos(x(9));
    t22 = t20*t21;
    t24 = -t14*t18-t14*t22;
    t25 = mb(3)*R;
    t26 = t3+Phi(3);
    t27 = cos(t26);
    t28 = sin(x(10));
    t29 = t27*t28;
    t31 = sin(t26);
    t32 = cos(x(10));
    t33 = t31*t32;
    t35 = -t25*t29-t25*t33;
    t40 = t2*t9;
    t42 = Omega*t6*x(3);
    t44 = t5*t10;
    t45 = x(3)^2;
    t48 = Omega^2;
    t49 = t16*t48;
    t52 = t14*t20;
    t54 = Omega*t17*x(4);
    t56 = t16*t21;
    t57 = x(4)^2;
```

```
t60 = t20*t48;
t63 = t14*t16;
t65 = Omega*t21*x(4);
t67 = t20*t17;
t71 = t48*e1;
t73 = t27*t48;
t76 = -v(1)*x(1)*abs(x(1))-c(1)*x(1)+2*t40*t42-t2*t44*t45+t14*t49*...
 t21-2*t52*t54+t14*t56*t57-t14*t60*t17+2*t63*t65-t14*t67*t57+mb(3)*...
 t27*t71+t25*t73*t32;
t77 = t25*t31;
t79 = Omega*t28*x(5);
t81 = t27*t32;
t82 = x(5)^2;
t85 = t31*t48;
t88 = t25*t27;
t90 = Omega*t32*x(5);
t92 = t31*t28;
t97 = t9*t48;
t100 = t2*t5;
t102 = Omega*t10*x(3);
t104 = t9*t6;
t107 = t5*t48;
t113 = -2*t77*t79+t25*t81*t82-t25*t85*t28+2*t88*t90-t25*t92*t82+mb(1)...
 *t9*t71+t2*t97*t6-2*t100*t102+t2*t104*t45-t2*t107*t10+mb(2)*t16...
 *t71-K(1)*x(6);
t118 = -t2*t44+t2*t104;
t121 = -t14*t67+t14*t56;
t124 = -t25*t92+t25*t81;
t146 = -v(2)*x(2)*abs(x(2))-K(2)*x(7)-c(2)*x(2)+mb(2)*t20*t71+t14*...
 t60*t21+2*t63*t54+t14*t22*t57+mb(3)*t31*t71+t14*t49*t17+2*t52*t65+...
 t14*t18*t57+t25*t85*t32;
t167 = 2*t88*t79+t25*t33*t82+t25*t73*t28+2*t77*t90+t25*t29*t82+mb(1)...
 *t5*t71+t2*t107*t6+2*t40*t102+t2*t7*t45+t2*t97*t10+2*t100*t42+t2...
 *t11*t45;
t169 = R^2;
t171 = x(8)^2;
t177 = Ks(1)*sign(x(8)-z);
t185 = Ks(1)*sign(x(8)+z);
t189 = e1*R;
t196 = -Kd(1)*t171*x(8)-Ke(1)*x(8)+t177*z/2-t177*x(8)/2-2*Vzeta(1)...
 *x(3)*abs(x(3))+t185*z/2+t185*x(8)/2-mb(1)*t48*t189*t10+u(1)...
 -Ks(1)*x(8)-Czeta(1)*x(3);
t203 = Ks(2)*sign(x(9)-z);
t208 = Ks(2)*sign(x(9)+z);
t217 = x(9)^2;
```

```
t222 = -2*Vzeta(2)*x(4)*abs(x(4))+t203*z/2-t203*x(9)/2+t208*z/2+t208...
 *x(9)/2-mb(2)*t48*t189*t17-Ks(2)*x(9)-Czeta(2)*x(4)-Ke(2)*x(9)-Kd(2)...
 *t217*x(9)+u(2);
t224 = x(10)^2;
t237 = Ks(3)*sign(x(10)-z);
t242 = Ks(3)*sign(x(10)+z);
t248 = -Kd(3)*t224*x(10)-Ke(3)*x(10)-mb(3)*t48*t189*t28-Ks(3)*x(10)...
 -2*Vzeta(3)*x(5)*abs(x(5))-t237*x(10)/2+t237*z/2+t242*z/2-Czeta(3)...
 *x(5)+u(3)+t242*x(10)/2;
B(1,1) = mb(1)+mb(2)+mb(3)+M(1);
B(1,2) = 0;
B(1,3) = t13;
B(1,4) = t24;
B(1,5) = t35;
B(1,6) = t76+t113;
B(2,1) = 0;
B(2,2) = M(2)+mb(1)+mb(2)+mb(3);
B(2,3) = t118;
B(2,4) = t121;
B(2,5) = t124;
B(2,6) = t146+t167;
B(3,1) = t13;
B(3,2) = t118;
B(3,3) = mb(1)*t169;
B(3,4) = 0;
B(3,5) = 0;
B(3,6) = t196;
B(4,1) = t24;
B(4,2) = t121;
B(4,3) = 0;
B(4,4) = mb(2)*t169;
B(4,5) = 0;
B(4,6) = t222;
B(5,1) = t35;
B(5,2) = t124;
B(5,3) = 0;
B(5,4) = 0;
B(5,5) = mb(3)*t169;
B(5,6) = t248;

% Calculate derivatives

[m,n]=size(B);
A1=B(:,1:n-1);
f1=B(:,n);
```

```
    sys=zeros(1,2*m);
    sys(1:5)=A1\f1;
    sys(6:10)=x(1:5);

% Output states

elseif abs(flag) == 3

    sys=x;

else

    sys = [];

end
```

APPENDIX E

MOVING BLOCK ANALYSIS CODE

The following group of MATLAB® programs can be used to perform a moving block modal damping analysis on a signals that are either unimodal or bimodal. The organization of the code is as follows:

mbloc is the primary code and calls **maxf2m** and **frecur** ; **maxf2m** calls **getmax** and **fft** (from the MATLAB® Signal Processing Toolbox function library) and **dft** . **dampA** is a separate code that is used to curve fit the resulting moving block plot.

MBLOC:

```
function [logF1,t1,logF2,t2,omega,N,Nb]=mbloc(X,sr)

% function [logF,t]=mbloc(X,sr)
%
% MBLOC calculates the magnitude of the discrete
% Fourier transforms of block segments of a signal
% for moving block damping analysis. This code is
% specifically designed to handle a signal with 1 or
% 2 dominant modes.
%
% X     -> vector which contains the signal
% sr    -> sampling rate at which signal was obtained
% logF1,2-> vector containing the natural logs of the
%          moving block function for each successive
%          block
% t1,2  -> vector containing the times initializing
%          each block
% omega -> frequency at which the moving block function
%          is evaluated
% N     -> signal length after it is padded with zeros
% Nb    -> block length.
% f     -> frequency spectrum of FFT (0 to Nyquist freq)
% Pxx   -> power spectrum (magnitude of FFT)
% w1,2  -> vector of frequencies over zoomed frequency
%          interval
%
% Copyright (c) 1997 by Chris S. Robinson
% All rights reserved
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Call routine which determines frequency of interest
% and block size for evaluation of moving block function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[omega,N,Nb,f,Pxx,w1,absF1,w2,absF2]=maxf2m(X,sr);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Pad signal with zeros if length is not a power of 2
% (this step is done because the fft routine contained
% in the function maxfum also pads the original signal
% with zeros if necessary). If signal length is a power
% of 2 initially then this step leaves the signal
% unaltered.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

l=N-length(X);
z=zeros(1,l);
X=[X z];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Evaluate the moving block function along signal using frecur and then fit
% resulting curve with linear least squares fit
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[logF1,t1]=frecur(X,omega(1),Nb(1),sr);
[logF2,t2]=frecur(X,omega(2),Nb(2),sr);
p1=polyfit(t1,logF1,1);
p2=polyfit(t2,logF2,1);
fit1=polyval(p1,t1);
fit2=polyval(p2,t2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot the FFT results, the zoomed DFT results and moving block function versus
% time
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

subplot(3,1,1)
plot(f,Pxx,'r');
xlabel('frequency (Hz)');
ylabel('radians^2/Hz');
title('Moving Block Plot');
subplot(3,1,2)
plot(w1,absF1,'r',w2,absF2,'g');
xlabel('frequency (Hz)');
```

```
ylabel('radians^2/Hz');
subplot(3,1,3)
plot(t1,logF1,'r',t1,fit1,'r--',t2,logF2,'g',t2,fit2,'g--');
xlabel('time (sec)');
ylabel('log(|F(w)|)');
grid
damp1=-p1(1)/(2*pi*omega(1))
damp2=-p2(1)/(2*pi*omega(2))
```

MAXF2M:

```
function [omega,N,NB,f,Pxx,w1,absF1,w2,absF2]=maxf2m(X,sr)

% function omega=maxf2m(X,sr)
%
% MAXF2N computes the 2 dominant maximum of a
% bi-modal signal, X,  in the frequency domain
% by using a fft for an initial estimate and then refining
% the solution by dividing the interval bounded by
% the nearest harmonics to the fft solution into subintervals.
% The Fourier coefficients are found at each of the
% frequencies defined by the subintervals, and a new
% maximum is found.  The intervals nearest to the maximum
% are further subdivided and the maximum obtained is
% considered an adequate estimate.
%
% X    -> Vector containing the values of the signal
% sr   -> sampling rate at which the signal generated/recorded
% omega -> the frequency of the dominant mode present in the
%         signal
% N    -> length of signal padded with zeros
% Nb   -> length of signal block that will be used for the
%         moving block analysis
% Pxx  -> Power spectrum of signal (magnitude of FFT)
% f    -> frequency spectrum of FFT
% u,l  -> two dominant frequencies estimated from FFT, refined
%         estimate will be made about these two frequencies
% absF1,2 -> Power spectrum over the zoomed intervals about
%           about the estimated frequencies
% omega1,2 ->The refined estimates of two dominant frequencies
% NB1,2   ->Block sizes, corresponding to the refined frequency
%           estimates, to be used for the moving block
%           calcualtions
%
% Copyright (c) 1997 by Chris S. Robinson
```

```
% All rights reserved

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine signal length and the number of points to be added
% to make that length a power of 2, then take the fast Fourier
% transform, instructing the fft routine to pad the signal with
% the proper number of zeros if necessary.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n=length(X);
N=2^(ceil(log2(n)));
XF=fft(X,N);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Take the results of the fft and determinethe power spectrum
% of the signal.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

m=length(XF);
Pxx=XF.*conj(XF)/m;
nyq=sr/2;
f=nyq*(0:N/2)/(N/2);
Pxx(N/2+2:N)=[];
Pxx(2:N/2)=2*Pxx(2:N/2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Get a first estimate of the signal frequencies by finding the
% frequencies corresponding to the spikes in the power
% spectrum, then take the largest two.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[maxV,indV]=getmax(Pxx);
[maxVs,i]=sort(maxV);
maxVs=fliplr(maxVs);
i=fliplr(i);
u=f(indV(i(1)));
l=f(indV(i(2)));
dOmega=abs(u-l);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Zoom in on estimated frequency and take discrete Fourier
% transform for frequencies on an interval around the initial
% estimates in order to refine the fft resolution
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
nb=floor(length(X)/2);
k1=nb/sr*u;
k2=nb/sr*l;
h=ceil(length(X)*dOmega/nyq)
Nb=nb-10*h:nb+10*h;
w1=Nb.^(-1)*k1*sr;
w2=Nb.^(-1)*k2*sr;
absF1=dft(X,sr,Nb,w1);
absF2=dft(X,sr,Nb,w2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine refined frequency estimate from results of discrete
% Fourier transforms of zoomed interval by finding frequency
% that corresponds to the maximum in the zoomed power spectrum
% plot.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[maxF1,ind1]=getmax(absF1);
[maxF2,ind2]=getmax(absF2);
[dummy,ind1a]=max(maxF1);
ind1b=ind1(ind1a);
[dummy,ind2a]=max(maxF2);
ind2b=ind2(ind2a);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Return the frequency and block size for two modes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

NB1=Nb(ind1b);
omega1=w1(ind1b);
NB2=Nb(ind2b);
omega2=w2(ind2b);
NB=[NB1 NB2];
omega=[omega1 omega2];
```

<u>FRECUR:</u>

```
function [logF,t]=frecur(X,omega,Nb,sr)

% function [logF,t]=frecur(X,omega,Nb,sr)
%
% FRECUR evaluates the moving block function for a signal
% given the frequency of interest and the block length
% using the recursion method.
```

```
% (only good for boxcar windowing)
%
% Copyright (c) 1997 by Chris S. Robinson
% All rights reserved


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Evaluate the Fourier coefficients for the initial block.
% This step also initializes the recursion formula for
% evaluation of all subsequent blocks.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 N=length(X);
 kb=omega*Nb/sr;
 Xb=X(1:Nb);
 c=cos(2*pi*kb/Nb*(0:Nb-1));
 s=sin(2*pi*kb/Nb*(0:Nb-1));
 a(1)=2/Nb*sum(Xb.*c);
 b(1)=2/Nb*sum(Xb.*s);
 t(1)=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Evalute the Fourier coefficients for the reamaining
% blocks by applying the recursion formula
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 for T=2:N-Nb;
   a(T)=(a(T-1)+2/Nb*(X(Nb+T-1)-X(T-1)))*cos(2*pi*kb/Nb)...
     +b(T-1)*sin(2*pi*kb/Nb);
   b(T)=-(a(T-1)+2/Nb*(X(Nb+T-1)-X(T-1)))*sin(2*pi*kb/Nb)...
     +b(T-1)*cos(2*pi*kb/Nb);
   t(T)=(T-1)/sr;
 end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Evaluate the natural log of the moving block function for
% each block.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 logF=1/2*log(a.^2+b.^2);
```

GETMAX:

```
function [maxV,indV]=getmax(X)
```

```
% function [maxV,indV]=getmax(X)
%
% GETMAX determines the relative maximum points
% in a vector of data (X).

n=length(X);
count=0;
for i=1:n-2
  a=X(i+1)-X(i);
  b=X(i+1)-X(i+2);
  if(sign(a) == sign(b) & a > 0)
    count=count+1;
    maxV(count)=X(i+1);
    indV(count)=i+1;
  end
end
```

DFT:

```
function absF=dft(X,sr,Nb,w)

% function absF=dft(X,sr,Nb,w)
%
% DFT computes the discrete Fourier transform
% magnitude of a signal, X, sampled at a rate, sr
% at the frequency w (w is in Hz).
%
% X   -> Vector containing signal
% sr  -> sampling rate at which signal was created
% Nb  -> Vector of number of points in sub-block of
%        signal over which the discrete Fourier
%        transform will be applied
% w   -> Vector of frequencies over which the discrete
%        Fourier transform will calculated (each has a
%        corresponding block size from the Nb vector)
%
% Copyright (c) 1997 by Chris S. Robinson
% All rights reserved

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Evaluate discrete Fourier Transform by calculating
% the Fourier coefficients at the frequency of interest.
% Return a vector containing the magnitudes of the dft
% at the frequencies contained in the vector w.
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:length(w)

  N=Nb(i);
  k=N*w(i)/sr;
  x=0:N-1;
  t1=2*pi*k*x/N;
  c=cos(t1);
  s=sin(t1);
  C=X(1:N).*c;
  S=X(1:N).*s;
  a=2/N*sum(C);
  b=2/N*sum(S);
  absF(i)=a^2+b^2;


end
```

DAMPA:

```
function y=dampA(logF,t,tstart,tstop,omega)

% function y=dampA(logF,t,tstart,tstop,omega)
%
% function DAMPA performs least squares fit of moving
% block data and calcualtes the damping modal damping
% from the slope of the fit.
%
% logF   -> vector of natural logs of the moving block
%           function values.
% t      -> time vector which corresponds to the times
%           at the beginning of the each block in the
%           moving block analysis.
% tstart -> Start time for section of moving block plot
%           to be least squares fitted.
% tstop  -> Stop time for section of moving block plot
%           to be least squares fitted.
% omega  -> frequency of used in moving block analysis.
%
% Copyright (c) 1997 by Chris S. Robinson
% All rights reserved
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Extract the pertinent section of the moving block plot
% according to the user defined start and stop times
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

u=ones(1,length(t));
ut1=tstart*u;
ut2=tstop*u;
[dummy,ind1]=min(abs(t-ut1));
[dummy,ind2]=min(abs(t-ut2));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Perform a first order polynomial fit to moving block plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

p=polyfit(t(ind1:ind2),logF(ind1:ind2),1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Use the resulting slope from the least squares fit to
% determine the modal damping and return this value
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

y=-p(1)/(2*pi*omega);
```

APPENDIX F

INPUT FILE FOR COMPLEX ROTOR-FUSELAGE MODEL

```
% This m-file serves as input file for running the simulink
% S-function helo3B.m.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Helo Physical and Aerodynamic parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Distance from fuselage center of mass to hub (length).

h=.7907;

% Hinge offset (length).

e1=.2791;

% Length of rotor blade (length).

R= 2.3809;

% Mass of rotor blades (mass).

mb(1)=0.01432;              % LBS/g (SLUGS)
mb(2)=0.01432;
mb(3)=0.01432;

% Effective mass of fuselage and moments of inertia (mass
% or mass*length^2).

M1=1.5486;
M2=1.3060;
I11=.13497;
I22=.4669;
I12=0;

% Blade azimuth phase angles (radians)

Phi(1)=0;
Phi(2)=2*pi/3;
Phi(3)=4*pi/3;

% Rotor speed (radians per/sec)
```

```
Omega=75.39;

% Blade spring stiffnesses

%     Linear springs for lead-lag (moment/radian)

    kl1(1)=143.85;
    kl1(2)=143.85;
    kl1(3)=143.85;

%     Linear springs for flap (moment/radian)

    kf1(1)=31.38;
    kf1(2)=31.38;
    kf1(3)=31.38;

%     Duffing springs for lead-lag (moment/radian^3)

    kl3(1)=0;
    kl3(2)=0;
    kl3(3)=0;

%     Duffing springs for flap (moment/radian^3)

    kf3(1)=0;
    kf3(2)=0;
    kf3(3)=0;

% Blade damping constants

%     Damping in lead-lag (moment/(rad/sec))

    cl(1)=3.55;
    cl(2)=3.55;
    cl(3)=3.55;

%     Damping in flap (moment/(rad/sec))

    cf(1)=0;
    cf(2)=0;
    cf(3)=0;

% Fuselage effective stiffness
```

```
%     Translational (force/length)

      KT1=3000;
      KT2=3000;

%     Rotational (moment/radian)

      KR1=280.35;
      KR2=46.60;

% Fuselage damping constants

%     Translational linear (force/(length/sec))

      CT1=0;
      CT2=0;

%     Translational nonlinear (force/(length/sec)^2)

      VT1=0;
      VT2=0;

%     Rotational linear (moment/(rad/sec))

      CR1=1.061449;
      CR2=1.29852;

%     Rotational nonlinear (moment/(rad/sec)^2)

      VR1=0;
      VR2=0;

% Aerodynamic parameters.

%     lift curve slope (1/radian)

      a=2*pi;

%     Parasite drag coefficient

      cd0=0.0079;

%     Air density (mass/length^3)

      rho1=0.002377;
```

```
%    Rotor chord (length)

     c=0.137;

%    Advance ratio

     mu=0;

%    Inflow ratio

     lambda=0;  % set to zero for all cases until trim
                % routine is setup

% Initial conditions


%    Fuselage translational rates (length/sec)

     xrtXi=0;
     xrtYi=0;

%    Fuselage rotational rates (radians/sec)

     xrrXi=.1;
     xrrYi=0;

%    Blade lead-lag rates (radians/sec)

     xr1li=0;
     xr2li=0;
     xr3li=0;

%    Blade flap rates (radians/sec)

     xr1fi=0;
     xr2fi=0;
     xr3fi=0;

%    Fuselage translational displacements (length)

     xtXi=0;
     xtYi=0;

%    Fuselage rotational displacements (radians)
```

```
    xrXi=0;
    xrYi=0;

%    Blade lead-lag displacement (radians)

    x1li=0;
    x2li=0;
    x3li=0;

%    Blade flap displacement (radians)

    x1fi=0;
    x2fi=0;
    x3fi=0;

% Form input matrices from above parameters

I1=[h,e1,R,mb(1),mb(2),mb(3),M1,M2,I11,I22,I12,...
   Omega,Phi(1),Phi(2),Phi(3)];

I2=[kl1(1),kl1(2),kl1(3),kl3(1),kl3(2),kl3(3);...
   kf1(1),kf1(2),kf1(3),kf3(1),kf3(2),kf3(3);...
   cf(1),cf(2),cf(3),cl(1),cl(2),cl(3)];

I3=[KT1,KT2,KR1,KR2;...
   CT1,CT2,CR1,CR2;...
   VT1,VT2,VR1,VR2];

I4=[a,cd0,rho1,c,mu,lambda];

I5=[xrtXi,xrtYi,xrrXi,xrrYi,xr1fi,xr2fi,xr3fi,xr1li,xr2li,xr3li];

I6=[xtXi,xtYi,xrXi,xrYi,x1fi,x2fi,x3fi,x1li,x2li,x3li];
```

REFERENCES

1. Tongue, B. H., "Limit Cycle Oscillations of a Nonlinear Rotorcraft Model," *AIAA Journal*, Vol. 22, July 1984, pp. 967-974.

2. Coleman, R. P., and Feingold, A. M., "Theory of Self Excited Mechanical Oscillations of Helicopter Rotors with Hinged Blades," NACA Report 1351, 1958.

3. Peters, D. A., Hohenemser, K. H., "Application of the Floquet Transition Matrix to Problems of Lifting Rotor Stability," *Journal of the American Helicopter Society*, April 1971, Vol. 16, pp. 25-33.

4. Hammond, C. E., "An Application of Floquet Theory to Prediction of Mechanical Instability," *Journal of the American Helicopter Society*, October 1974, pp.14-23.

5. Ormiston, R. A., "Rotor-Fuselage Dynamics of Helicopter Air and Ground Resonance", *Journal of the American Helicopter Society*, April 1991, pp. 3-20.

6. Venkatesan, C. and Friedmann, P., "Aeroelastic Effects in Multi-Rotor Vehicles With Application to Hybrid Heavy Lift System, Part I: Formulation of Equations of Motion," NASA Contractor Report 3822, 1984.

7. Warmbrodt, W., Freidmann P., "Formulation of Coupled Rotor/Fuselage Equations of Motion,", *Vertica*, Vol. 3, pp. 245-271.

8. Tongue, B. H., Flowers, G., "Nonlinear Rotorcraft Analysis," *International Journal of Nonlinear Mechanics*, Vol. 23, No 3, September 1987, pp. 189-203.

9. Tongue, B. H., Flowers, G., "Nonlinear Rotorcraft Analysis Using Symbolic Manipulation," *Applied Mathematical Modeling*, Vol. 12, April 1988, pp.154-160.

10. Tongue, B. H., and Jankowski, M. D., "Construction and Analysis of a Simplified Nonlinear Ground Resonance Model," *Journal of Sound and Vibration*, 1988, 122, pp. 233-241.

11. Tang, D. M., and Dowell, E. H., "Influence of Nonlinear Blade Damping on Helicopter Ground Resonance Instability," *Journal of Aircraft*, Vol. 23, No 2, February 1986, pp. 104-110.

12. Straub, F. K., "Study to Eliminate Ground Resonance Using Active Controls," NASA Contractor Report 166609, 1984.

13. Straub, F. K., Warmbrodt, W. "The Use of Active Controls to Augment Rotor/Fuselage Stability," *Journal of the American Helicopter Society*, July 1985, 13-21.

14. Ormiston, R. A., "The Challenge of The Damperless Rotor.", Paper No. 68, 22nd European Rotorcraft Forum, Brighton, U.K. September 1996.

15. SIMULINK® *Dynamic Simulation Software User's Guide*, The MathWorks, Inc. ,1992.

16. Nakamura, S., *Applied Numerical Methods with Software*, Prentice Hall, 1991.

17. Bramwell, A. R. S., *Helicopter Dynamics,* Edward Arnold Publishers Ltd., London, 1976.

18. Hammond, C. E., and Doggett, R. V. , "Determination of Subcritical Damping by Moving-Block/Randomdec Applications," *NASA Symposium on Flutter Testing Techniques*, NASA SP-415, Oct. 1975.

19. Bousman, W. G, and Winkler, D. J., "Application of the Moving Block Technique", *Proceedings of the 22nd Annual AIAA Structural Dynamics and Materials Conference*, 81-0653CP, April 1981.

20. Kaza, K. R. V., and Kvarternik R. G., "Examination of Flap-Lag Stability of Rigid Articulated Rotor Blades", *Journal of Aircraft,* Vol. 16, December 1979.

21. Deutsch, M. L., "Ground Vibrations of Helicopters," *Journal of the Aeronautical Sciences,* Vol. 15, No. 5, May 1946, pp. 223-228.

22. Takashi, M. D., Friedmann, P. P., "A Simple Active Controller to Suppress Helicopter Air Resonance in Hover and Forward Flight," AIAA 29th Structures Structural Dynamics and Materials Conference, Williamsburg, VA, April 18-20, 1988, Paper No. AIAA-88-2407.

23. Weller, W. H., "Fuselage State Feedback for Aeromechanical Stability Augmentation of a Bearingless Main Rotor," *Journal of the American Helicopter Society*, April 1996, pp. 85-93.

24. Wood, R. E., Powers, R. W., Cline, J. H., and Hammond, C. E., "On Developing and Flight Testing a Higher Harmonic Control System," *Journal of the American Helicopter Society*, Vol. 30, No. 1, January, 1985, pp. 3-20.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, Ste 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Chairman, Code AA/Co 1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, CA 93943-5000

4. Professor E. Roberts Wood, Code AA/Wd 2
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California 93943-5000

5. Professor D. A. Danielson, Code MA/Dn 1
   Department of Mathematics
   Naval Postgraduate School
   Monterey, California 93943-5000

6. Professor Clyde W. Scandrett, Code MA/Sc 1
   Department of Mathematics
   Naval Postgraduate School
   Monterey, California 93943-5000

7. LT Christopher S. Robinson, USN 1
   1318 Wakefield Dr.
   Virginia Beach, Virginia 23455

8. LT Rob King, USN, Code AA/Ki 1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California 93943-5000

9. Dr. Friedrich K. Straub 1
McDonnell Douglas Helicopter Systems
5000 E. McDowell Road
Mesa, AZ 85215

10. Professor Peretz P. Friedman 1
46-147N Eng. IV
University of California
Los Angeles, CA 90024-1597

11. Dr. Robert A. Ormiston 1
U. S. Army Aeroflightdynamics Dir.
MS 215-1
NASA Ames Research Center
Moffet Field, CA 94035

12 . Mr. William G. Bousman 1
U. S. Army Aeroflightdynamics Dir.
MS 215-1
NASA Ames Research Center
Moffett Field, CA 94035

13. Dr. C. Eugene Hammond 1
100 Cove Colony Road
Maitland, FL 32751

14. Dr. William Warbrodt 1
M/S T12-B
NASA Ames Research Center
Moffet Field, CA 94035

15. Professor David A. Peters 1
Box 1185
Dept. of Mechanical Engineering
Washington University
St. Louis, MO 63130

16. Mr. William Weller 1
11 Pine Road
West Hartford, CT 06119

17. Dr. John Shaw 1
655 Newlin Road
Springfield, PA 19064

18. Mr. W. Euan Hooper 1
729 Rhoads Drive
Springfield, PA 19064-1609

19. Dr. Jing G. Yen 1
1812 Lakemont Court
Arlington, TX 76013

20. Mr. William D. Anderson 1
780 Charleston Drive
Roswell, GA 30075-2885

21. Mr. Wen-Liu Miaio 1
10 Crossbow Lane
Easton, CT 06612

22. Dr. Robert G. Loewy 1
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332

23. Mr. C. Thomas Snyder 1
Bldg. N207-1
NASA Ames Research Center
Moffet Field, CA 94035

24. Dr. Dev Banerjee 1
McDonnell Douglas Helicopter Systems
5000 E. McDowell Road
Mesa, AZ 85215